

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication :

2 253 434

(A n'utiliser que pour les
commandes de reproduction).

A5

BREVET D'INVENTION *

(21)

N° 73 42707

(54)

Ordinateur à mémoire virtuelle.

(51)

Classification internationale (Int. Cl.²).

G 06 F 15/00.

(22)

Date de dépôt

30 novembre 1973, à 14 h 37 mn.

(33) (32) (31)

Priorité revendiquée :

(47)

Date de la mise à la disposition du
public du brevet

B.O.P.I. — «Listes» n. 26 du 27-6-1975.

(71)

Déposant : COMPAGNIE HONEYWELL BULL, 94, avenue Gambetta, 75020 Paris.

(72)

Invention de : Claude Carré, Guy Hiot, Jean Michel et Claude Rolland.

(73)

Titulaire : *Idem* (71)

(74)

Mandataire :

* La présente publication n'a pas été précédée d'une publication de la demande correspondante.

U

Vente des fascicules à l'IMPRIMERIE NATIONALE, 27, rue de la Convention — 75732 PARIS CEDEX 15

L'invention concerne le domaine des ordinateurs de grande puissance et plus particulièrement un ordinateur à mémoire virtuelle. Plus spécifiquement, l'invention se rapporte à une organisation de gestion de mémoire virtuelle pour un tel ordinateur.

5 Un ordinateur à mémoire virtuelle comporte, d'une manière générale, une mémoire principale à accès sélectif, associée en général à une mémoire tampon rapide, et une mémoire auxiliaire de grande capacité qui renferme les informations, souvent très volumineuses, qui ne sont pas immédiatement nécessaires pour le traitement en cours d'exécution par l'ordinateur.

10 Aux informations contenues dans la mémoire principale et dans la mémoire auxiliaire sont associées des adresses logiques. Des moyens de traduction sont prévus pour retrouver l'adresse physique des informations (dans la mémoire principale ou auxiliaire) à partir de leur adresse logique et des moyens sont prévus pour transférer automatiquement en mémoire principale
15 les informations contenues dans la mémoire auxiliaire nécessaires pour l'exécution du traitement en cours par l'ordinateur.

Le but recherché par l'adressage virtuel est ainsi de faire apparaître à l'utilisateur le système de mémoire virtuelle de l'ordinateur comme une mémoire unique ayant la capacité de la mémoire auxiliaire mais une rapidité
20 presque égale à celle de la mémoire principale.

La présente invention a pour objet une organisation de mémoire virtuelle en vue d'assurer la gestion de ladite mémoire virtuelle de telle sorte qu'elle mette à la disposition du processeur central les informations nécessaires au traitement en cours tout en ne renvoyant de la mémoire principale à la
25 mémoire auxiliaire les informations nécessaires à d'autres traitements qui doivent être exécutés ensuite par le processeur, que dans la mesure strictement nécessaire au traitement en cours et quivant un ordre de priorité de façon à réduire au minimum les transferts entre les dispositifs de mémoire et les pertes de temps qui résultent.

30 A cette fin, la dimension du plus grand espace d'adresses vides de la mémoire principale est mise en mémoire dans une table des communications de façon à permettre de déterminer immédiatement s'il existe un espace suffisant dans la mémoire principale pour recevoir le groupe d'informations appelé "segment" qui doit être chargé dans la mémoire principale. Dans l'affirmative, une
35 série de comparaisons entre la dimension de l'espace que le segment doit occuper et celles des espaces vides permet de charger le segment dans un espace vide approprié.

Dans la négative, il est fait à nouveau recours à la table des communications qui contient également la dimension du plus grand espace disponible formé par la combinaison d'espaces d'adresses occupés par des
40

segments qui ont été désignés par leur propriétaire comme susceptibles d'être renvoyés dans la mémoire auxiliaire, et des espaces vides voisins. Si cet espace libre combiné est supérieur à la dimension du segment à charger, une nouvelle étape de comparaison permet de charger ce segment dans un espace approprié. Sinon, une recherche est alors effectuée pour trouver un segment occupant dans la mémoire principale un espace convenable (éventuellement en combinaison avec les espaces libres voisins) pour recevoir le segment à charger et dont la valeur de priorité permet son renvoi en mémoire auxiliaire.

- 10 Si un tel espace est trouvé, le segment occupant cet espace est éventuellement renvoyé dans la mémoire auxiliaire si sa copie n'y est plus validé tandis que le nouveau segment est chargé à sa place dans la mémoire principale. Sinon, une opération générale de renvoi des segments en mémoire auxiliaire sur la base de leur ancienneté et de compaction des espaces vides de la mémoire principale est effectuée pour libérer l'espace de mémoire principale nécessaire.

- 20 Selon une autre caractéristique de l'invention, chaque groupe de processus d'un programme d'utilisateur déclare, lors de son initialisation, la quantité d'espace qu'il doit occuper dans la mémoire principale et des moyens sont prévus pour calculer à tout instant l'espace de mémoire principale effectivement occupé par ledit groupe de processus. Le rapport entre la quantité d'espace déclarée et la quantité effectivement utilisée est alors utilisé pour le classement dudit groupe de processus. D'une manière générale, le groupe de processus qui a dépassé la quantité déclarée se voit sanctionné (par exemple, s'il demande à nouveau un espace supplémentaire et que la mémoire principale ne dispose pas d'un tel espace libre, seules les informations appartenant audit groupe peuvent être extraites de la mémoire principale pour libérer l'espace nécessaire) tandis qu'un groupe de processus peut, dans certain cas, se voir garantir le droit à l'utilisation de la totalité de l'espace déclaré dans la mémoire principale.

- 30 Selon encore une autre caractéristique de l'invention, le temps passé à gérer les échanges entre la mémoire principale et la mémoire auxiliaire est constamment contrôlé et lorsqu'un seuil d'emballlement indicatif d'un risque de pertes de performances est atteint, le groupe de processus le moins prioritaire est extrait de la mémoire virtuelle tandis que le chargement de nouveaux groupes de processus est arrêté jusqu'à ce que la fraction de temps passée à gérer les échanges soit redevenue normale.

- 35 Selon encore une autre caractéristique de l'invention, la mémoire auxiliaire est divisée en blocs ayant tous la même longueur et les blocs de mémoire auxiliaire sont attribués un à un aux divers groupes de processus.

BAD ORIGINAL

A chaque groupe de processus sont associés des moyens de mémoire pour conserver les adresses réelles desdits blocs et l'adresse, à l'intérieur du bloc en cours de remplissage, du début de l'espace non encore occupé. Un nouveau bloc est attribué au groupe de processus chaque fois
5 que son bloc en cours a été complètement rempli. Les blocs ont, de préférence, une dimension supérieure à la longueur d'une série successive d'informations ou segment du groupe de processus de sorte qu'ils reçoivent normalement un ou plusieurs segments.

La table des communications contient notamment un répertoire des
10 blocs attribués et un compte des blocs disponibles qui sont mis à jour chaque fois qu'un nouveau bloc est attribué et chaque fois qu'un groupe de processus est terminé de façon à détecter à l'avance l'état de saturation de la mémoire auxiliaire et à remettre immédiatement à la disposition des autres groupes de processus les blocs libérés.

15 Grâce à ces dispositions, prises séparément ou en combinaison, et à d'autres dispositions et caractéristiques de l'invention qui ressortiront de la description plus détaillée qui va suivre, la mémoire virtuelle peut être gérée d'une manière extrêmement souple avec une occupation maximale de l'espace de mémoire disponible et une organisation optimale de la mémoire
20 principale.

Dans les dessins annexés donnés uniquement à titre d'exemple :

- la figure 1 est une représentation générale, sous forme d'un schéma-bloc, de l'organisation d'un ordinateur à mémoire virtuelle mettant en oeuvre les principes de l'invention;
- 25 - la figure 2 est un schéma bloc d'un ordinateur typique dans lequel l'invention peut être mise en oeuvre de façon particulièrement avantageuse;
- la figure 3 est une représentation schématique de la mémoire principale d'un ordinateur selon l'invention ;
- 30 - la figure 4 représente schématiquement la mémoire principale et les diverses tables utilisées pour l'adressage par la base du système de diverses structures d'informations (groupes de processus, sémaphores, pile d'attente des processus);
- la figure 5 représente le format d'une adresse effective et les tables
35 utilisées pour localiser à partir de cette adresse un segment et l'information recherchée à l'intérieur dudit segment;
- la figure 6 représente, sous forme de schéma-bloc, les diverses interfaces entre les composantes du complexe de gestion de mémoire virtuelles selon l'invention;
- 40 - la figure 7A représente la matrice complète des primitives

demanderes et appelées dans le cadre du complexe de gestion de mémoire virtuelle;

- la figure 7B représente graphiquement les diverses interfaces énumérées dans la matrice de la Fig. 7A;

5 - la figure 8 représente graphiquement l'état de la mémoire principale dans des conditions d'utilisation typique, les informations relatives à ladite mémoire contenues dans une "table des communications" étant indiquées par leurs noms mnémoniques;

- la figure 9 représente une en-tête typique de segment;

10 - la figure 10 représente une en-tête typique de trou;

- la figure 11 représente le format d'un descripteur de segment indirect;

- la figure 12 représente le format d'un descripteur de segment direct;

15 - la figure 13 est une matrice indiquant la signification de la combinaison des divers bits indicateurs des descripteurs de segment;

- la figure 14 est un schéma montrant l'adressage par un descripteur direct du segment correspondant;

20 - les figures 15A à 15E représentent l'organisation de la partie de la mémoire principale réservée aux utilisateurs. La figure 15A montre les trous, pseudo-trous et trous virtuels qui peuvent être détectés dans ladite partie de mémoire; les figures 15B à 15E représentent les liens créés entre les segments et les trous par les différents mots de leur entête et qui permettent la détermination de ces trous, pseudo-trous, et trous virtuels; la figure 15B représente les liens entre les segments et trous successifs (mot 1, 2 des en-tête), la

25 figure 15D représente les liens entre les segments éligibles pour être échangés (mots 3, 4 des en-tête des segments éligibles) et la fig. 15E représente les liens entre les segments non éligibles pour être échangés (mots 3, 4 des segments non éligibles).

30 - la figure 16 est un organigramme de la fonction d'attribution de la mémoire des utilisateurs;

- les figures 17A et 17B constituent, en combinaison, un organigramme de la fonction de changement d'attribution de la mémoire principale;

35 - les fig. 18A à 18I constituent, en combinaison, un organigramme de la fonction de déchargement des segments de la mémoire principale;

- la figure 19 est un organigramme de la fonction de détection et de suppression de l'emballement des échanges entre la mémoire principale et la mémoire auxiliaire.

1 - Introduction

40 Sur la figure 1 on a représenté un schéma de principe de la gestion

de mémoire virtuelle selon l'invention.

Un processeur central 10 communique avec la mémoire virtuelle 20 en adressant la mémoire 20 au moyen d'adresses logiques qui sont transmises à un système de gestion de la mémoire 60 lequel décode les adresses logiques en adresses physiques pour adresser la mémoire principale 30, si l'information désirée se trouve dans la dite mémoire principale, ou pour commander un contrôleur d'Entrée/Sortie 50, entre la mémoire principale 30 et une mémoire de masse 40, si l'information demandée se trouve dans la mémoire de masse 40, afin de la transférer à la mémoire principale 30. Ensuite, l'information adressée est transmise au Processeur 10. Le système de gestion de mémoire communique avec une série de sources d'informations à savoir des registres généraux 61, une table des communications 62, les en-têtes des segments et des trous 63, les descripteurs des segments 64, les segments de commandes des groupes de processus 65, la table d'activité des travaux 66, la base du système 67 et les sémaphores du système 68 qui fournissent et reçoivent les informations nécessaires au système de gestion de la mémoire virtuelle non seulement pour assurer le décodage des adresses logiques et le transfert des informations requises par le Processeur, de la mémoire auxiliaire à la mémoire principale, mais également pour assurer toujours les autres fonctions de gestion automatique de la mémoire virtuelle, indépendamment du processeur central, telles que la recherche d'espace disponible dans la mémoire principale, le renvoi en mémoire auxiliaire d'informations non utilisées, la surveillance de la saturation des mémoires, de l'emballement des échanges entre les mémoires etc...

Cette brève description donne un aperçu de la complexité de la gestion d'une mémoire virtuelle dans un ordinateur de grande puissance. Pour cette raison il est désirable de donner tout d'abord un exposé du moins succinct de l'architecture générale d'un système typique de traitement des données de grande puissance dans lequel les principes de la présente invention peuvent être avantageusement utilisés. Après cette description préliminaire, on pourra effectuer une description plus détaillée des techniques de gestion de mémoire virtuelle qui font l'objet de cette invention.

L'organisation générale de la description qui va suivre est donnée dans la table des matières, à la fin de la présente description.

35 2 - Matériel de traitement

Les organes matériels principaux d'un système d'ordinateur typique de la catégorie dans laquelle la présente invention peut être avantageusement utilisée sont représentés sur la figure 2 des dessins. Un tel système comporte un Processeur Central représenté d'une manière générale dans le rectangle en traits interrompus désigné par la référence 100, une

mémoire principale 101, un contrôleur d'Entrée/Sortie (CES) 102 et une mémoire auxiliaire (réalisée sous la forme d'une mémoire à disques ou à tambour) 103.

5 Les opérations détaillées qui sont effectuées dans le Processeur central 100 sont commandées en grande partie (mais, comme on le verra, pas complètement) par des microprogrammes mis en mémoire dans une unité de mémoire de commande 106. Une unité de mémoire de commande de ce type est décrite dans le brevet des EUA n° 3.634.883 délivré le 11 janvier 1972 au nom de Léonard L. Kreidermacher et un exposé supplémentaire des techniques de commande par microprogrammation peut être trouvé dans "Microprogramming : Principles and Practices" de Samir S. Husson, Prentice Hall, Inc. (1970).

15 Les mémoires de commande sont habituellement des dispositifs de mémoire inaltérable à semi-conducteurs qui sont capables d'avoir des temps de cycle très rapides, chaque emplacement de stockage de la mémoire de commande servant à commander un cycle ou une partie de cycle du Processeur Central 100.

20 Lorsque chaque emplacement de stockage de la mémoire de commande est lu, son contenu est décodé pour conditionner le fonctionnement d'autres unités fonctionnelles du Processeur Central. L'unité de mémoire de commande 106 communique avec l'unité de gestion des données 107, l'unité de chargement d'instructions 108, l'unité de commande d'adresses 109 et l'unité arithmétique et logique 110, par l'interface 111 de la mémoire de commande. L'interface 111 comporte une logique pour la modification d'adresse de la mémoire de commande, les tests, le contrôle d'erreur et la génération d'adresses par le "matériel".

30 Dans la disposition représentée sur la fig. 2, une mémoire tampon 113 est utilisée pour conserver temporairement les informations qui ont été le plus récemment extraites de la mémoire principale 101 et qui sont utilisées au cours du traitement par le Processeur Central. La mémoire tampon 113 est caractérisée par sa grande rapidité et par sa capacité relativement faible (par rapport à la mémoire principale 101) et est divisée logiquement en blocs adressables sans équivoque possible. Ces blocs sont appelés pages, chaque page de la mémoire tampon 113 étant adressée en examinant le contenu du répertoire de la mémoire tampon 115. La mémoire tampon 113 et le répertoire 115 permettent, par leur combinaison, d'obtenir une amélioration importante de la vitesse de fonctionnement globale, en particulier pour les traitements de données comportant de nombreuses itérations et au cours desquels il se produit des accès fréquents à des données qui ont été traitées récemment, du fait de l'exécution répétitives d'un petit

40

nombre d'instructions.

L'unité de gestion des données 107 est utilisée pour transférer l'information désirée de la mémoire aux registres du Processeur Central, suivant le découpage et aux moments appropriés et pour réaliser le masquage
5 au cours des opérations d'écriture partielle.

L'unité de chargement d'instructions 108 fournit les instructions aux autres organes du Processeur Central et a pour objectif de rendre l'instruction suivante disponible avant achèvement de l'exécution de l'instruction en cours. Pour qu'elle puisse remplir cette fonction, l'unité 108 comporte un
10 registre d'instructions (non représenté) qui contient normalement plusieurs instructions, et en utilisant les informations de la mémoire de commande 106, elle demande des instructions de la mémoire principale 101 (ou de la mémoire tampon 113) avant que l'instruction soit effectivement requise, maintenant ainsi son registre d'instructions constamment à jour. Les instructions
15 sont ainsi normalement préalablement chargées dans le registre d'instructions au cours des cycles de mémoire non utilisés afin d'accélérer le débit d'instructions, donnant ainsi à l'avance au Processeur Central des informations sur la longueur et le format de ces instructions qui doivent être exécutées dans un futur rapproché.

L'unité de commande d'adresses 109, qui fonctionne également sous la commande d'un microcode de l'unité de mémoire de commande 106 des unités de commande est responsable du développement d'adresse dans le Processeur Central 100. L'unité 109 comporte une mémoire associative (non représentée) qui met en mémoire les adresses de base des segments de mémoire les plus
25 récemment utilisés, avec leur numéro de segment respectif. (L'adressage segmenté sera décrit ultérieurement de façon plus détaillée). Chaque fois qu'une demande d'accès à la mémoire est présentée, le contenu de la mémoire associative est - examiné pour déterminer si le développement d'adresse pour l'adresse demandée a déjà été effectué et pour déterminer alors les
30 résultats de ce développement précédent, mis en mémoire. Dans l'affirmative, le contenu de la mémoire associative de l'unité de commande d'adresses 109 peut être utilisé directement et un développement d'adresse inutile est évité. Lorsque, cependant, la mémoire associative ne contient pas une adresse déjà développée, l'unité 109 interroge les tables d'adresses de la mémoire principale
35 afin de développer l'adresse physique réelle dans la mémoire principale de l'information recherchée.

L'unité arithmétique et logique (UAL) 110, qui fonctionne sous la commande d'un microcode de l'unité de mémoire de commande 106, est la zone de travail principale du Processeur Central et exécute les opérations arithmétiques et les manipulations de données requises par l'instruction. Une unité
40

de mémoire locale 118 coopère avec l'unité arithmétique et logique 110 et sert à mettre en mémoire l'information de commande et à mettre momentanément en mémoire les facteurs et les résultats partiels engendrés au cours de la manipulation des données.

- 5 Enfin, le Processeur Central 100 comporte une série de registres de commande de processus 120 dont le contenu définit l'état du Processeur après l'exécution de chaque instruction. Ces registres de commande de Processus comportent (1) des registres de base, (2) des registres généraux, (3) des registres scientifiques, (4) le compteur d'instructions, (5) des
10 registres d'état et (6) le registre T (registre de pile):

3 - Structures d'informations visibles pour le "Matériel"

3.1 - Introduction à l'adressage segmenté

- Avant de considérer la structure des informations qui peuvent être directement adressées par le matériel de traitement, il paraît nécessaire de
15 définir le concept d'adressage segmenté. Considéré d'une manière générale, les segments sont des blocs d'instructions (appelés "Procédures") ou des blocs de données dont on laisse les dimensions croître ou se réduire au cours de l'exécution d'un processus. A chaque segment est associé un descripteur qui se trouve dans une petite mémoire supplémentaire appelée
20 "table de segments" mettant en mémoire les caractéristiques distinctives ou "attributs" des segments. Lorsqu'un segment est chargé dans la mémoire principale physique, son adresse physique ou absolue est enregistrée dans l'un des répertoires des tables de segments. La table des segments est alors utilisée pour associer le nom symbolique (ou adresse logique) d'un segment
25 donné à son adresse physique réelle. Ainsi, tout "mot" individuel dans un segment donné peut être référencé par une adresse logique de la forme (STN, D) dans laquelle STN (entrée de table de segments) identifie l'entrée de la table de segments qui donne l'adresse physique du premier mot du segment dans la mémoire physique et D est le déplacement ou décalage dans
30 le segment particulier qui identifie le mot particulier référencé. La technique de l'adressage d'un emplacement effectif dans la mémoire physique par l'intermédiaire d'un répertoire (du type table de segments) est appelée adressage indirect.

- Il est possible d'effectuer un adressage indirect à plusieurs niveaux.
35 Par exemple, la mémoire principale peut contenir un certain nombre de tables de segments, chacune pointant les adresses d'un certain nombre de segments. Ces multiples tables de segments peuvent elles-mêmes être pointées par un répertoire "maître" (appelé "tableau" de mots de tables de segments). A ce point, il peut exister une nouvelle hiérarchie d'adressage
40 indirect suivant laquelle il existe plusieurs tableaux chacun étant pointé par

un répertoire encore plus général.

Le concept de segmentation et d'adressage indirect est essentiel dans les techniques de gestion de mémoire qui font l'objet de la présente invention et sera décrit plus en détail ci-après. A ce point de la description, cependant, l'explication donnée ci-dessus est suffisante pour permettre l'examen des structures d'informations mises en mémoire dans la mémoire principale accessibles au "matériel".

3.2. Organisation des commandes et plan du système d'exploitation dans la mémoire principale.

On rappellera qu'un microcode utilisé pour commander les opérations du matériel est mis en mémoire dans la mémoire de commande 106 représentée sur la Fig. 2. Cependant, la quantité totale d'informations de microprogrammation peut s'étendre au-delà de la capacité de la mémoire de commande 106, et le dépassement de capacité peut être placé dans la mémoire principale en commençant à l'adresse physique 0 et en allant jusqu'à une "adresse limite" dont la valeur est dictée par l'importance globale de la microprogrammation nécessaire pour un système et une configuration donnés.

Le plan des structures d'informations "visibles pour le matériel" à des adresses physiques prédéterminées est représenté sur la figure 3. L'espace d'adresse 201 qui s'étend de l'adresse physique 0 à l'adresse limite est appelé "zone de matériel" et est réservé à l'information de microprogrammation supplémentaire est chargée dans la mémoire principale au moment du chargement initial du système, est utilisée par le Processeur Central, par le contrôleur d'entrée/sortie et par d'autres organes matériels remplissant des fonctions d'exécution, mais il est "invisible" à la fois pour l'utilisateur et pour le système d'exploitation.

En outre la zone du matériel 201 est protégée contre les modifications provoquées par l'utilisateur et les processus du système d'exploitation de deux manières : (1) en lui donnant un caractère absolu (2) par une interruption d'exception. A l'achèvement de l'établissement de la microprogrammation détaillée du matériel, le nombre d'adresses nécessaire pour la zone du matériel dans la mémoire principale est compté de sorte que, lorsque cette zone est chargée dans la mémoire principale, en commençant à l'adresse physique 0, le numéro de l'adresse suivante disponible peut être déterminé. Cette adresse appelée "adresse limite" a une valeur qui est écrite dans la région du matériel 201 par le réalisateur du système à un emplacement prédéterminé (par exemple adresse physique 10). En conséquence, au cours du processus de chargement initial du système d'ordinateur, le matériel de traitement peut avoir accès à cet emplacement physique prédéterminé et

l'adresse limite peut être chargée dans un registre prévu à cet effet ou registre d'adresse limite (BAR). On dit alors que la zone de matériel est mise en mémoire au-dessous du "BAR" (c'est-à-dire à des adresses inférieures à la valeur contenue dans le registre "BAR") tandis que le système d'exploitation et la programmation de l'utilisateur sont mis en mémoire au-dessus du BAR.

Le système d'adressage segmenté est alors avantageusement utilisé pour isoler la zone du matériel 201 des autres processus. Lorsque le système d'exploitation et la programmation de l'utilisateur sont chargés, les tables de segments donnant les adresses physiques sont modifiées en ajoutant la valeur de l'adresse limite en tant que valeur de décalage à tous les pointeurs de segments initiaux. De cette manière, l'adresse limite apparaît comme un 0 physique à la fois au système d'exploitation et aux processus des utilisateurs. Il devient ainsi impossible aussi bien au système d'exploitation qu'aux processus des utilisateurs d'avoir accès à l'information située au-dessous de l'adresse limite ou d'en modifier le contenu.

A titre de sécurité supplémentaire (en cas d'erreur fonctionnelle au cours du traitement), un autre contrôle de sécurité est utilisé. Les microprogrammes du matériel sont identifiés et se distinguent des codes du système d'exploitation et des utilisateurs par des bits d'entête d'identification. Les vérifications peuvent ainsi être effectuées pour engendrer une interruption d'exception des procédures du système qui sert à empêcher la modification de la zone du matériel et à avertir le personnel d'exploitation de telles difficultés, lorsqu'elles se rencontrent.

Après achèvement du processus de chargement initial, le système est dans un état où il peut effectuer le travail demandé par l'utilisateur. En général, le système d'exploitation a été chargé, les dispositifs d'entrée/sortie ont été vérifiés et mis à l'état de fonctionnement, et les ressources de mémoire disponibles pour l'utilisateur ont été calculées, les résultats étant mis en mémoire dans des structures d'informations adressables. Plus spécifiquement, l'état du système dans le régime statique (dans l'attente de la mise en route par l'opérateur par enfoncement d'une touche de clavier du pupitre de commande pour mettre en route un processus du système) peut être qualifié par (1) le contenu de la mémoire principale, (2) les fichiers de la mémoire auxiliaire qui sont ouverts et accessibles, (3) les fonctions du système d'exploitation qui sont disponibles et (4) l'état des dispositifs spécifiques qui permettent à un utilisateur d'introduire un ordre de traitement.

Les structures d'informations implantées dans la mémoire principale physique à ce moment peuvent être considérées comme résidant dans les six zones distinctes 201 à 206, telles que représentées sur la figure 3. La zone réservée au matériel de traitement 201 (déjà décrite) s'étend de l'emplacement

0 à l'adresse limite. Ensuite, la "base du système" 202 appartenant au système d'exploitation s'étend dans l'espace d'adresses commençant à l'adresse BAR. Les informations mises en mémoire dans la base du système ont un format prédéterminé (qui sera examiné de façon plus détaillée en se référant à la

5 figure 4) et sont directement adressables par le système d'exploitation. D'autres structures d'informations (adressables indirectement par des pointeurs dans la base du système) occupent la zone 203. Ces structures (qui seront examinées plus en détail ultérieurement) comportent les tables G..... et J ainsi que les tables des dispositifs logiques et des services physiques. L'es-

10 pace d'adresses 203, en combinaison avec la base du système 202, constituent ce qu'on appelle "l'extension de la base du système".

Ensuite, par ordre d'adresses physiques croissantes, on trouve la zone de mémoire du système d'exploitation résidant 204 qui contient en permanence les segments du système d'exploitation résidant en mémoire principale

15 (à la fois les procédures et les données), les structures permanentes de commande de processus du système d'exploitation et les tables de gestion de mémoire virtuelle. Les structures d'informations dans la zone de mémoire résidante 204 sont maintenues en permanence dans la mémoire principale pour

ou
une/plusieurs des raisons ci-après.

- 20
- 1) elles sont utilisées à une grande fréquence;
 - 2) elles traitent les erreurs machine ou les interruptions;
 - 3) elles traitent les transferts entre la mémoire auxiliaire et la mémoire principale créant ainsi le cadre nécessaire au système de mémoire virtuelle.

25 Les tables de gestion de mémoire virtuelle sont contenues dans la zone de mémoire du système résidant 204 et comportent les types d'informations ci-après :

- 1) l'adresse absolue de la fin du Noyau (le noyau est la partie du système résidant en mémoire principale qui s'étend à partir de l'adresse limite

30 jusqu'à la fin de la zone de mémoire du système résidant 204;

- 2) l'adresse absolue de la fin de la mémoire de commande (la zone de mémoire de commande 205 contient les segments de commande du système d'exploitation et ceux des groupes de processus des utilisateurs qui seront examinés plus en détail en se référant à la fig. 4).
- 35 3) le volume de mémoire échangeable disponible maximal (la mémoire échangeable est la partie restante 206 de la mémoire principale qui contient les segments des procédures et des données qui peuvent être renvoyés à la mémoire auxiliaire);
- 4) la mémoire de commande actuellement disponible;

40 5) l'adresse de début de la mémoire auxiliaire;

- 6) l'adresse de fin de la mémoire auxiliaire;
- 7) le volume de mémoire actuellement disponible dans la mémoire auxiliaire.

4 - Structures d'informations visibles pour l'utilisateur

4.1 - Classification des procédures et processus des utilisateurs

Jusqu'à ce point, la description d'ensemble du système de traitement a été effectuée dans l'optique principale du matériel de traitement. On a commencé par décrire le matériel de traitement et on a conclu en donnant une explication des structures d'informations visibles pour le matériel de traitement, telles qu'elles sont implantées dans les emplacements de mémoire. A ce point, il est utile de s'arrêter et de considérer le système du point de vue de l'utilisateur, principalement en vue de définir certains termes et de parvenir à une meilleure compréhension des besoins de l'utilisateur.

L'utilisateur prépare son problème pour qu'il soit résolu par la machine par étapes et les étapes peuvent être plus facilement distinguées en définissant les termes ci-après : Procédure, programme, unité de compilation et modules de liaison (de chargement).

Une procédure est un jeu d'instructions, de constantes et d'instructions de commande écrites à l'origine par un programmeur dans un langage de programmation dont elles respectent les règles. Bien que écrite initialement à l'état de programme d'origine, une procédure peut exister dans d'autres états, par exemple à l'état de programme d'origine développé, compilé et chargé (en modules). Dans le dernier de ces états, une procédure est conforme aux règles du système d'exploitation en ce qui concerne son format, sa syntaxe et son aspect.

Un programme contient au moins une procédure principale. Il peut contenir d'autres procédures et il contient en outre des informations d'identification telles que le nom du programmeur, le langage d'origine utilisé, le compilateur utilisé etc.... Un programme peut aussi être considéré comme un ensemble de procédures écrites dans un langage spécifique auquel sont ajoutées des informations supplémentaires qui identifient l'ensemble spécifique et commandent l'utilisation de l'ensemble des procédures.

Une unité de compilation est une structure d'informations créée par un compilateur à partir d'un programme d'origine (ou d'un programme d'origine développé). Cette structure d'informations existe dans un format standard compatible avec le système d'exploitation et contient une traduction des instructions d'origine en termes d'instructions du matériel de représentations de données et d'informations de commande compatibles avec le système d'exploitation. Une unité de compilation peut être organisée en ce qu'on appelle "segments provisoires" c'est-à-dire que le compilateur prend des

décisions de groupage sur la manière de diviser le code, les constantes et les informations de commande en segments. De tels segments provisoires restent cependant incomplets du fait que les numéros de segments définitifs ne leur ont pas été encore attribués et, dans certains cas, du fait

5 que ces segments provisoires sont à l'état de simple squelette. Les unités de compilation peuvent être réunies en bibliothèque.

Un module de liaison (ou module de chargement) est une structure d'informations également en format standard créée par les processus du système d'exploitation en utilisant les unités de compilation. Un module de

10 liaison diffère d'une unité de compilation du fait que des numéros de segments ont été attribués de façon définitive à ses segments et qu'il peut contenir des segments supplémentaires provenant d'autres unités de compilation. Il est important de noter que le module de liaison comporte des structures de commande (l'une de ces structures particulièrement importante étant un

15 segment appelé segment de commande de groupe de processus (PGCS)) qui sont presque complètement développées. La tâche la plus importante à exécuter avant que les procédures contenues dans un module de liaison puisse être exécutées est l'achèvement de la résolution d'adresses faisant intervenir des entités contenues dans le noyau du système d'exploitation. Cette

20 tâche restante est exécutée au moment où le module est chargé dans la mémoire virtuelle pour créer un groupe de processus.

L'existence de structures d'informations, telles que les modules de liaison, les unités de compilation, les procédures etc... doit être distingué de l'exécution des opérations de traitement des données utilisant ces struc-

25 tures. L'exécution effective du travail de traitement des données peut être divisée en travaux, étapes de travail et processus.

Un travail est une unité principale du système. Il est défini comme étant un ensemble comportant une ou plusieurs étapes de travail qui sont liées entre elles du point de vue de l'utilisateur pour des raisons logiques.

30 Par exemple, un utilisateur peut désirer rassembler un groupe d'étapes de travail pour des travaux de comptabilité et récapitulation. Dans d'autres situations, l'exécution de certaines étapes de travail peut dépendre de l'exécution d'autres étapes de travail ou certains fichiers doivent passer d'une étape de travail à une autre sans qu'il soit possible à une troisième

35 étape ne faisant pas partie du travail de s'introduire entre temps et de modifier le fichier. Du point de vue du système en conséquence, un travail est une unité de travail programmable dans lequel la programmation est basée sur des paramètres qui sont principalement fonction des besoins de l'utilisateur.

40 Une étape est une unité de travail secondaire et est entité à laquelle

les ressources (mémoire auxiliaire, mémoire secondaire, dispositifs, fichier mémoire principale, etc...) sont attribuées. Lorsqu'une étape doit être exécutée par le système, un nom de module de chargement est identifié avec une liste des ressources nécessaires. En d'autres termes, une étape

5 est une demande d'exécution d'un module de chargement dans un cadre de ressources donné afin d'accomplir le travail.

Lorsque le système reçoit une demande d'exécution d'une étape (la demande étant établie dans un "langage de commande de travail" standardisé) le système d'exploitation entreprend tout d'abord la tâche d'attribuer des ressources à la demande au fur et à mesure que ces ressources deviennent disponibles, puis d'initialiser les structures d'informations requises pour l'exécution avant de commander la procédure initiale de l'étape. Avant que la commande soit lancée, cependant, tous les segments du module de chargement sont chargés en mémoire virtuelle (c'est-à-dire sont

15 soit dans la mémoire principale soit dans le dispositif d'échange coopérant) et certains de ces segments peuvent avoir été mis à jour pour un grand nombre de raisons. Egalement, avant que l'exécution effective commence, une structure de commande d'étape est construite dans un fichier secondaire du système pour les informations accédées peut fréquemment et éventuellement volumineuses qui ne sont pas nécessaires dans la mémoire virtuelle.

20 La combinaison du module chargé dans la mémoire virtuelle et mis à jour et de cette structure de commande d'étape placée dans un fichier de mémoire secondaire est appelée un groupe de processus.

Un processus est l'exécution commandée d'instructions sans concurrence. Plus précisément, un processus est une unité de commande à l'intérieur d'une séquence d'instructions, une entité abstraite se déplaçant à l'intérieur de l'instruction d'une procédure au fur et à mesure que la procédure est exécutée par un Processeur. Les attributs d'un processus au cours d'exécution varient ainsi continuellement avec le temps.

Dans le système calculateur examiné ici, le processus peut exister dans l'un des sept états qui peuvent être détectés par le matériel de traitement du système. Par le terme "détecté" on entend que le matériel de traitement exécute diverses instructions et procédures de microprogrammation pour réaliser la distribution des tâches, les changements d'état et pour

30 maintenir les structures d'informations qui définissent l'état dans lequel se trouve le processus. Il existe sept états de processus possibles et ceux-ci sont représentés par une zone de code dans le bloc de commande du processus (PCB), une structure de commande contenue dans la mémoire principale. Ces sept états sont les suivants :

40 (1) en cours d'exécution : lorsque le processus est affecté au

Processeur central et que l'état actuel du processus est décrit dans les registres de commande de processus 120 (fig.) du matériel. Dans les autres états, l'état du processus est décrit dans son bloc de commande de processus ;

- 5 (2) prêt : dans ce cas, le processus est dans une file d'attente en attendant d'être affecté au Processeur Central;
- (3) en attente : le processus est dans une file d'attente en attendant une action des sémaphores (les sémaphores seront décrits ultérieurement);
- 10 (4) suspendu pendant l'attente : le processus, alors qu'il est en attente, a été expressément suspendu par l'action d'un autre processus exécutant une instruction de "SUSPENSION";
- (5) suspendu à l'état prêt : le processus a été suspendu alors qu'il est à l'état prêt;
- 15 (6) suspendu après exception : le processus a été suspendu par le matériel à la suite de la détection d'une conduction d'exception (des exemples d'exception seront décrits ultérieurement); ou
- (7) suspendu avant en mise en route : (auquel cas le processus n'a jamais été exécuté).
- 20 Plusieurs groupes de processus peuvent normalement être chargés en même temps en mémoire virtuelle. Lorsque l'état en cours d'exécution d'un processus donné est arrêté pour une raison quelconque (par exemple du fait de la nécessité d'exécuter une opération d'entrée/sortie avant que le processus puisse continuer), le Processeur Central peut poursuivre avec
- 25 un processus différent. L'état du processus quittant l'état en cours d'exécution (tel que représenté dans les registres de commande de processus du Processeur Central) est enregistré dans le bloc de commande de processus (PCB) dudit processus tandis que l'information du bloc de commande de processus (PCB) relative au nouveau processus est transférée aux registres
- 30 de commande du processus du Processeur Central de sorte que ce processus peut être réactivé (ou repris au point où il s'est arrêté lorsqu'il a cessé d'être en cours pour la dernière fois).

4.2 - Structures adressées à partir de la base du système

- A chaque processus correspond un bloc de commande de processus
- 35 (PCB) qui est accessible au matériel par l'intermédiaire d'un mécanisme d'adresses nécessitant l'intervention de la base du système 202 et de deux tables de pointeurs, la table J et la table P, représentées sur la Fig.4.

La table J désignée par la référence 211 sur la fig. 4 est pointée par le mot de table J qui occupe l'espace d'adresse s'étendant de l'adresse

- 40 limite à l'adresse limite plus quatre (BAR + 4) dans la mémoire physique.

Le mot de table J comporte un pointeur (c'est-à-dire l'adresse absolue) désignant le début de la table J (qui se trouve dans la mémoire physique, dans la zone d'extension 203 de la base du système représentée sur la Fig. 3). La table J peut typiquement contenir jusqu'à 255 entrées, chaque entrée pointant
5 une table P unique désignée par la référence 213 sur la Fig. 4. Chaque table P peut à son tour comporter typiquement jusqu'à 255 entrées permettant ainsi un maximum de 65.025 processus qui peuvent être "connus" de façon unique et non équivoque par le système.

Chaque entrée de la table P pointe un bloc de commande de processus
10 (PCB) du type de celui désigné par la référence 215 sur la Fig. 4. Les blocs de commande de processus se trouvent à l'intérieur de segments de commande de processus qui se trouvent eux-mêmes dans la mémoire de commande (zone 205 Fig. 3).

A la suite du mot de table J, dans la base du système, se trouve le
15 mot de table G. Le mot de table G pointe, par l'intermédiaire de la table G 217, un groupe de segments G dont le premier est le segment Go 219 et dont les autres sont représentés par le segment représentatif G221. La table G 217 et les segments G 219 et 221 se trouvent dans la zone d'extension de la base du système 203 (Fig. 3) de la mémoire principale. A la suite du mot de
20 table G, une série de mots d'exception du système 0 à 7 sont écrits dans le système de base suivis par un mot d'exception d'horloge interne et d'un mot d'exception de canal, ces mots étant inscrits respectivement entre les adresses physiques BAR + 8 et BAR + 48. Chacun de ces mots contient un numéro de segment G et un déplacement (G, D) pour adresser un mot particulier dans
25 les segments G. Chaque mot d'exception de la base du système pointe ou désigne ainsi une variable de synchronisation détectée par le matériel appelé un "sémaphore", utilisée pour établir une communication entre deux processus exécutés concurremment mais de façon asynchrone.

Un sémaphore couple ainsi dynamiquement des processus asynchrones
30 par la combinaison de deux opérations du matériel, utilisant les variables mises en mémoire à une adresse donnée de sémaphore dans un segment G. La première opération, appelée opération P (p-op) fonctionne comme demande de message qui est placée à l'adresse de sémaphore et reste à cette adresse si un message n'a pas été précédemment laissé par une opération V (V-op).
35 En signifiant exagérément, la combinaison d'un code P-op et d'une adresse de sémaphore est une requête pour déterminer si un message a ou non été laissé par un autre processus exécutant une opération V(V-op). L'opération V-op comporte à la fois l'adresse de sémaphore où le message doit être laissé et le message lui-même.

40 Les opérations V-op et P-op peuvent se produire dans un ordre

quelconque. Si une opération P-op est exécutée en premier lieu et qu'une seconde opération P-op arrive avant que n'arrive une opération V-op portant un message les demandes de message P-op sont placées dans une file d'attente pour attendre l'arrivée du premier message V-op. De même, si plusieurs messages V-op arrivent avant toute demande P-op à l'adresse du sémaphore, les messages sont placés dans une file d'attente de messages.

Les demandes de message P-op et les messages V-op sont appareillés soit sur la base d'un ordre de priorité, soit sur la base du principe dernier entré-premier sorti, soit encore sur la base du principe premier entré-premier sorti et les résultats obtenus sont placés dans un registre général prédéterminé où ils peuvent être utilisés par le système d'exploitation pour exécuter le traitement de façon continue.

Ainsi, les sémaphores peuvent être utilisés pour remplacer des "interruptions" utilisées dans certains systèmes d'exploitation. Par exemple, un processus qui est en cours d'exécution peut être placé à l'état suspendu lorsqu'il nécessite une opération d'Entrée/Sortie. L'exécution du processus est suspendue dans le Processeur Central et une P-op est placée à une adresse prédéterminée de sémaphore à ce moment. Après achèvement de l'opération d'Entrée/Sortie, le processus d'entrée/sortie transmet un message V-op à cette adresse de sémaphore indiquant que l'opération d'entrée/sortie a été achevée, satisfaisant la demande de message P-op et retirant le processus de l'état d'attente et le plaçant à l'état prêt (plaçant le processus dans la file d'attente des processus prêts).

La file d'attente des processus prêts (PR/Prêts) est elle-même une structure d'information qui se trouve dans le segment GO qui est pointé par la base du système par une adresse double G, D qui se trouve; comme représenté sur la fig. 4, dans la file d'attente interne du Processeur, à l'emplacement BAR + 48 de la base du système. L'attribution du Processeur à un processus en attente est effectuée sous la commande du matériel qui utilise un système de file d'attente avec priorité sur la base premier entré, premier sorti à l'intérieur de la priorité.

4.3 - Adressage à partir des blocs de commande de groupe de processus.

Comme précédemment indiqué en se référant à la fig. 4, l'adresse d'un bloc de commande de processus (PCB) pour un processus donné est développée en s'adressant tout d'abord au mot de table J visible pour le matériel, situé au début de la structure d'information de base du système contenue dans la mémoire principale. Le mot de table J contient l'adresse de début (ou de base) de la table J. Le numéro J d'un processus particulier identifie une entrée particulière de la table J et l'entrée de la table contient l'adresse

de base d'une table **P** particulière. Le numéro **P** attribué à un processus donné identifie une entrée particulière de la table **P** et cette entrée de la table **P** indique le bloc de commande de processus (PCB) (c'est-à-dire contient l'adresse de ce bloc dans la mémoire physique) d'un processus particulier.

Ainsi, un développement d'adresse important est effectué uniquement pour localiser la structure d'information dans la mémoire de commande qui décrit le processus à exécuter. Cette structure, le bloc de commande de processus (PCB) est alors utilisée comme point de départ d'une nouvelle structure d'informations utilisée pour attribuer, manipuler et accéder aux instructions et données effectives utilisées dans la procédure de l'utilisateur.

L'adresse absolue des informations requises par un processus par exemple, un facteur de données appelé par une instruction donnée, est développée à partir d'une adresse logique ou effective qui fournit l'information nécessaire pour exécuter une série d'opérations de consultations de tables qui sont représentées d'une manière simplifiée sur la figure 5.

L'adresse effective (qui n'est pas l'adresse absolue physique) d'un mot donné qui doit être extrait de la mémoire principale est désignée sur la figure 5 par la référence générale 216. Cette adresse comporte deux parties principales à savoir un numéro de segment et un groupe de bits de déplacement. La partie de l'adresse effective 216 contenant le numéro de segment est elle-même divisée en un bit unique de tableau (désigné **Y** sur la figure 5), un groupe de bits constituant un numéro de table de segments (STN) et un troisième groupe de bits constituant un numéro d'entrée de table de segments (STE).

On doit également rappeler que l'adresse effective est elle-même le produit d'un processus préexistant de développement d'adresse exécuté par le système d'exploitation sur la base de l'instruction initiale en langage d'origine (écrit par un programmeur) qui a été convertie par le compilateur en un formant compatible avec le système et à laquelle des numéros de segments ont été attribués au cours du processus de création du module de chargement et que cette adresse comporte des bits de déplacement modifiés résultant du processus de combinaison d'un numéro de déplacement dans le module de chargement avec des déplacements (éventuellement négatifs) provenant des registres de base ou des registres d'indexage. Des détails du procédé et des dispositifs utilisés pour développer l'adresse effective 216 sont décrits dans la demande de brevet n°7317765 du 16. 5. 73 au nom de la demanderesse.

Une étude plus générale du développement d'adresses effectives est

donnée dans "The Multics System : An Examination of its Structure" par Eliot I. Organick, the MIT Press (1972).

Sans entrer dans les détails, le processus de développement de l'adresse 216 à partir de l'information contenue dans une instruction d'une
5 procédure de module de chargement est le suivant . L'instruction comporte une syllabe adresse qui spécifie une adresse effective en désignant un registre de base (et éventuellement également un registre d'indexage) et un déplacement à l'intérieur du segment. Le registre de base spécifié (typiquement huit registres de base sont disponibles pour chaque processus)
10 contient une adresse segmentée constituée par le numéro de table de segments STN, un index désignant le poste de table de segments STE et un déplacement D dans le système désigné. Le développement d'adresse consiste alors à localiser le segment référencé dans le registre de base (STN, STE) et à effectuer le développement du déplacement à l'intérieur du segment
15 par la compilation additive du contenu de déplacement dans la syllabe adresse, du contenu de déplacement dans le registre de base, et s'il est référencé, du contenu de déplacement dans le registre d'indexage.

Comme précédemment indiqué en se référant à la fig. 4, le bloc de commande de processus, désigné dans son ensemble par la référence 215
20 sur la Fig. 5, est situé physiquement dans la zone de mémoire de commande de la mémoire principale et a une adresse qui peut être déterminée par le matériel en accédant séquentiellement au mot de table J dans la base du système, à la table J et à une table P. Le bloc de commande de processus 215 contient notamment deux mots d'espace d'adresses 216 et 217. Le bit
25 Y dans l'adresse effective 216 étant soit un zéro soit un Un identifie celui des mots d'espace d'adresses qui doit être utilisé dans le développement d'adresse.

Comme représenté sur la Fig. 5, le bit Y dans l'adresse effective 216 est supposé être un 0 désignant ainsi le mot 0 d'espace d'adresses 216
30 dans le bloc PCB 215. Le mot 216 contient l'adresse absolue de la base (adresse de début) d'un tableau de mots de tables de segment 220.

Il existe deux tableaux de ce type dans la mémoire principale, l'un qui est utilisé pour développer les adresses de "grands" segments et l'autre pour adresser les "petits" segments. Les grands segments contiennent
35 typiquement un maximum de 4 millions de multiplats tandis que les petits segments contiennent un maximum de 65.000 multiplats.

Le tableau de mots de tables de segments 220 contient un certain nombre de mots de tables de segments, le mot particulier recherché étant localisé au moyen du numéro de table de segments (STN) dans l'adresse effective
40 216. La valeur du numéro STN identifie le déplacement ou décalage dans le

tableau 220 qui localise le mot de table de segments (STN) recherché, dans ce cas le mot de table de segment 221.

Chaque mot de table de segments contient notamment l'adresse absolue de la base d'une table de segments du type représenté et désigné par la référence 225. La table de segments 225 contient un certain nombre de descripteurs de segments et le descripteur recherché 227 est identifié au moyen de la valeur de l'entrée de table de segments (STE) dans l'adresse effective 216. Par exemple si la valeur de l'entrée STE est 12, le douzième descripteur de la table 225 est identifié.

Le descripteur de segment 227 contient notamment l'adresse absolue de la base d'un segment 230. L'emplacement de la mémoire principale contenant le mot adressé est alors spécifié par les bits de déplacement D dans l'adresse effective 216.

A ce point de la description, plusieurs observations importantes peuvent être faites.

En premier lieu, les segments dont le segment de mémoire 230 est un exemple sont des unités utilisés collectivement par les processus. Par exemple, si deux processus ou davantage nécessitent un accès à un même bloc de données, des copies de ce bloc de données n'ont pas besoin d'exister du fait que les tables de segments dans différents groupes de processus peuvent être organisés pour comporter des descripteurs de segments désignant un bloc unique. Ainsi, les informations mises en mémoire dans une unique zone de mémoire peuvent exister dans l'espace d'adresses de plusieurs groupes de processus, ce qui permet ainsi d'économiser une partie importante d'espace de mémoire.

En second lieu, les segments sont également des unités "d'échange" dans le cadre d'un système de mémoire virtuelle. Comme on l'examinera plus en détail ultérieurement, chaque descripteur de segment contient, en plus de l'adresse de base du segment qu'il désigne, un bit de présence "p" indiquant si le segment réside ou non dans la mémoire principale. S'il n'est pas présent les procédures de gestion de mémoire virtuelle répondent en déplaçant automatiquement le segment de la mémoire auxiliaire à la mémoire principale et le descripteur du segment est alors modifié pour indiquer l'adresse de la mémoire principale attribuée au segment.

Un segment qui n'est plus nécessaire peut également être sorti en bloc de la mémoire principale et à ce moment le bit de présence P de son descripteur est modifié et son adresse dans la mémoire principale est remplacée par sa nouvelle adresse dans la mémoire auxiliaire.

Les procédures du système d'exploitation peuvent prendre la forme de segments "partagés". Une fonction du système d'exploitation est dite

"partagée" lorsqu'elle est effectuée par les procédures du système d'exécution sous la commande du processus de l'utilisateur, c'est-à-dire si les procédures sont pointées par le bloc de commande de processus (PCB) de l'utilisateur. Les fonctions sont dites "non partagées" si elles sont exécutées par
5 les procédures du système d'exécution sous la commande d'un processus du système. C'est la nature collective des segments qui permet le placement des structures d'informations du système d'exploitation dans l'espace d'adresses du processus des utilisateurs.

Les avantages présentés par la mise en oeuvre des fonctions du système d'exploitation sous forme d'un ensemble de segments qui fait partie de
10 l'espace d'adresses des processus qui ont besoin de ces fonctions (au lieu de les rattacher à un processus unique du système) sont les suivants :

En premier lieu, le passage d'un processus à un autre implique des moyens support généraux de gestion de processus importants. Par exemple,
15 les registres de commande de processus du Processeur Central doivent être vidés dans un bloc de commande de processus (PCB) et chargés du contenu du bloc PCB du processus suivant avant de continuer. Dans le cas, cependant, où les segments du système d'exploitation se trouvent dans l'espace d'adresses du processus qui demande à être desservi, les communications entre
20 le processus de l'utilisateur et le processus du système sont considérablement simplifiées. En second lieu, chaque fois qu'il existe plusieurs demandes pour un processus du système d'exploitation émanant de plusieurs processus d'utilisateurs, le parallélisme peut être réalisé au cours de l'exécution de la fonction d'une manière très naturelle sans programmation compliquée
25 (ce qui serait le cas avec des machines commandées par interruptions) et sans le coût (en espace mémoire) rencontré dans un système où la mémoire de l'utilisateur est segmentée mais où la mémoire du système d'exploitation ne l'est pas.

Certains attributs du descripteur de segment doivent également être
30 notés. En premier lieu, les segments ne se chevauchent pas et un segment donné est décrit par un seul et même descripteur de segment (bien qu'une table de segments puisse être pointée par une pluralité de processus différents). Il en résulte que le caractère collectif est étendu au niveau des tables de segments.

Etant donné que plusieurs descripteurs sont rassemblés dans une seule
35 table de segments, les types de tous les segments rattachés à une table spécifique doivent être identiques. En d'autres termes, le descripteur d'un grand segment ne peut être placé dans une table contenant les descripteurs de petits segments. En outre étant donné qu'il n'existe qu'un seul descripteur de
40 segment pour décrire un segment, tous les processus utilisant collectivement

un segment ont les mêmes droits d'accès à ce segment.

Il est suffisant de mettre à jour l'information contenue dans les descripteurs de segments uniquement au moment où les segments sont déplacés de la mémoire principale à la mémoire auxiliaire au cours des opérations d'échange de la mémoire virtuelle. En particulier, il n'est pas nécessaire de maintenir une liste de longueur variable du processus utilisant collectivement un segment à un moment donné. En outre, étant donné que tout segment d'utilisation collective a le même numéro dans tous les processus, il n'est pas besoin de prévoir un segment spécial de liens rattaché à chaque processus.

4.4 - Protection Interprocessus

Il est essentiel dans tout système comportant plusieurs utilisateurs de protéger les programmes des utilisateurs les uns vis à vis des autres et de protéger le système d'exploitation vis à vis des procédures des utilisateurs. Cette protection est obtenue à la fois par la segmentation et par un "système d'anneaux".

La segmentation isole de façon inhérente les espaces d'adresses des différents programmes (qui lors de l'exécution sont matérialisés sous forme de groupes de processus). Ceci découle du fait qu'il est impossible à un groupe de processus de référencer, soit accidentellement soit volontairement, une entité appartenant à un autre groupe de processus c'est-à-dire une entité qui ne se trouve pas dans l'espace d'adresses du processus en cours d'exécution mais qui au contraire se trouve dans l'espace d'adresses d'un autre groupe de processus. Le seul recouvrement d'espace d'adresses qui existe résulte des segments qui sont partagés notamment les segments "publics" qui sont accessibles à tous les utilisateurs et sont chargés lors de la création du système.

Cependant, la segmentation à elle seule n'assure pas une protection convenable. Il est à l'évidence indésirable de permettre la modification des segments publics du système d'exploitation par les procédures écrites par l'utilisateur même si, comme expliqué ci-dessus, de tels segments publics existent dans l'espace d'adresses du processus de l'utilisateur. En outre, l'utilisateur peut lui-même désirer désigner certaines procédures comme étant privilégiées c'est-à-dire comme n'étant pas modifiables par des procédures moins privilégiées. Pour cette raison, un système de protection par anneaux est utilisé pour créer une hiérarchie des niveaux de protection dans laquelle

les procédures peuvent être classées. Exprimé d'une manière simplifiée, un segment contenant des procédures se voit attribuer une valeur d'anneau permettant à cette procédure d'appeler (ou de désigner) tout segment de procédure ou de données ayant la même valeur d'anneau ou une valeur d'anneau supérieure. Inversement, une procédure ayant une valeur d'anneau donnée n'a pas la possibilité de désigner ou d'appeler des segments ayant une valeur d'anneau

inférieure plus privilégiée. Ainsi, le créateur d'un segment, en attribuant une valeur d'anneau appropriée, a la possibilité de permettre son utilisation collective et en outre de spécifier le type d'utilisation permis. Une explication détaillée du matériel et des procédures utilisés pour le contrôle par anneaux apparaît inutile pour la compréhension de la présente invention et ne sera donc pas donnée ici. Des détails de ce système sont donnés dans la demande de brevet N° 7342706 du même jour au nom de la demanderesse intitulée : "Protection de l'information dans un système de multitraitement de l'information mettant en oeuvre un concept d'anneaux pour représenter les différents niveaux de privilèges entre processus".

5 - Gestion de la mémoire virtuelle

5.1 - Description générale

Une description générale des principales caractéristiques d'un système typique de traitement des données à grande puissance ayant ainsi été effectuée, on décrira maintenant de façon plus spécifique les aspects du système qui intéressent plus directement l'invention. En premier lieu, on donnera une description générale de la gestion de mémoire virtuelle puis on présentera une description des techniques spécifiques de gestion de mémoire qui font l'objet de la présente invention.

Les principes servant de base à la conception d'un système de gestion de mémoire virtuelle peuvent être décrits en se référant aux trois fonctions principales du système: la gestion des segments, la gestion de la mémoire principale et la gestion de la mémoire auxiliaire.

5-1.1 Gestion des segments

La "composante" de gestion des segments constitue la commande "superviseur" essentielle du système de gestion de mémoire virtuelle et peut être appelée explicitement (par la procédure écrite par l'utilisateur) ou implicitement (par le biais des exceptions). Les autres composantes fonctionnelles du système de gestion de mémoire virtuelle servent de support à la gestion des segments.

Toutes les composantes de gestion des segments sont "partagées" (c'est-à-dire qu'elles constituent des procédures publiques placées dans les espaces d'adresses de tous les processus des utilisateurs) mais l'utilisation simultanée de ces procédures est évitée par l'utilisation de sémaphores qui sont verrouillés à la suite de l'entrée du processus requérant et libérés lors du retour.

La gestion des segments fait appel aux procédures de gestion de la mémoire principale et de la mémoire auxiliaire afin de charger des segments dans la mémoire principale. Les segments qui doivent être renvoyés dans la mémoire auxiliaire ne sont pas immédiatement échangés, cependant, mais sont simplement placés dans une liste de segments qui sont éligibles pour

être échangés (ESW LIST) comme on le verra, les segments éligibles pour être échangés (segments ESW) ne sont effectivement écrits en mémoire auxiliaire que lorsqu'ils contiennent des données modifiées (c'est-à-dire lorsqu'il n'existe pas déjà une copie équivalente dans la mémoire auxiliaire).

5 5-1.2 - Gestion de la mémoire principale

La gestion de la mémoire principale est le nom donné à cette structure d'informations principale au moyen de laquelle la mémoire principale peut être utilisée en "multiplex". Cette structure fait appel aux listes des adresses des segments chargés dans la mémoire physique et aux listes des adresses des "trous" (adresses de mémoire non utilisées) à la fois dans la mémoire de commande et dans la mémoire échangeable (zones 205 et 206, fig.3) de la mémoire principale physique.

La zone de mémoire de commande 205 est chargée en commençant à la fin de la zone de mémoire du système d'exploitation résidant 204 (fig.3) par ordre d'adresses physiques croissantes tandis que la zone de mémoire des segments échangeables 206 (fig.3) est chargée à partir de la limite supérieure de la mémoire principale par ordre d'adresses physiques décroissantes.

La limite entre la mémoire de commande et la mémoire échangeable, établie lors du chargement initial du système, peut être déplacée pour accroître la mémoire de commande ou la mémoire échangeable, selon les besoins.

Une fois que la mémoire échangeable a été complètement remplie, l'espace est libéré pour des segments actifs supplémentaires en retournant des segments inactifs à la mémoire auxiliaire conformément à des critères de remplacement prédéterminés. Finalement, la "mémoire virtuelle" (constituée par la combinaison de la mémoire principale et de la mémoire auxiliaire) peut atteindre une condition du système dite condition "d'emballement" dans laquelle l'activité E/S associée aux échanges entre les mémoires est considérée excessive. L'une des fonctions de la gestion de la mémoire virtuelle est, en conséquence, la détection de cet état d'"emballement" et l'exécution de procédures en vue de l'éliminer (par exemple en empêchant l'ordonnement de nouveaux groupes de processus ou en empêchant les activités d'un groupe de processus peu prioritaire).

Les procédures de la gestion de la mémoire principale sont initialement entrées à partir des procédures de gestion des segments.

35 5-1.3 - Gestion de la mémoire auxiliaire

La "composante" de gestion de la mémoire auxiliaire dans le "complexe" de gestion de mémoire virtuelle remplit les fonctions demandées par la gestion des segments notamment l'attribution et la libération de la mémoire auxiliaire sur la base d'un compte des groupes de processus, la lecture et l'écriture rapides simultanément de segments entiers et l'optimisation de l'occupation

de l'espace de la mémoire auxiliaire. De même que la gestion des segments et la gestion de la mémoire principale, la gestion de la mémoire auxiliaire est une structure d'informations collective qui réside en mémoire principale dans le noyau du système.

- 5 La mémoire auxiliaire elle-même est de préférence réalisée sous la forme d'une partie d'une mémoire à disques. Un tel dispositif de mémoire échangeable comporte une pluralité de têtes de lecture/écriture positionnées pour explorer une pluralité de pistes concentriques sur chacun d'une pluralité de disques tournant à grande vitesse sur un arbre commun. Ainsi, l'adresse d'un bloc de données quelconque dans la mémoire auxiliaire peut être spécifiée en identifiant le cylindre (c'est-à-dire la série de pistes sur plusieurs disques ayant le même rayon) la tête (identifiant celle des pistes qui doit être lue) et l'adresse relative (la position angulaire du point de départ du bloc de données). Bien que d'autres systèmes d'adressage de disques ou de tam-
- 10 bours puissent être utilisés, le format cylindre tête et adresse relative (C, H, R) est utilisé ici, à titre d'exemple typique.

- Les segments ont leur espace réservé dans la mémoire auxiliaire sur la base d'un compte de groupe de processus. En d'autres termes, lorsqu'un groupe de processus est chargé, un espace est attribué à chacun de ses segments, à ce moment, dans la mémoire virtuelle. A l'achèvement du groupe de processus, l'espace réservé dans la mémoire auxiliaire est libéré et rendu aux autres utilisateurs. La fonction de gestion de la mémoire auxiliaire sert à attribuer et à libérer la mémoire auxiliaire ainsi qu'à construire le programme du canal physique (PCP) nécessaire lorsqu'un segment
- 20 est chargé ou déchargé.

- L'espace d'adresses physiques de la mémoire auxiliaire est divisé en blocs attribuables d'égales dimensions, chaque cylindre de la mémoire auxiliaire contenant normalement un nombre entier de ces blocs et chaque bloc étant à son tour, composé d'un nombre entier de pistes. Les dimensions des blocs attribuables ainsi que toutes les autres caractéristiques du fichier de la mémoire auxiliaire sont des paramètres déterminés au moment de la création du système qui sont contenus dans la table de communications (COM-TAB) qui sera examinée ultérieurement au paragraphe 5.4.2.
- 30

- Dans les cas usuels, les segments ont des dimensions sensiblement plus petites qu'un bloc assignable unitaire d'où il résulte que les segments peuvent normalement être lus ou écrits avec un seul positionnement des têtes et avec un programme de canal relativement simple. Le fichier de la mémoire auxiliaire est ainsi organisé pour permettre, dans les cas usuels, l'attribution d'une partie de la piste ou des pistes consécutives sur un même cylindre à un segment désiré afin de pouvoir lire ou écrire un segment entier
- 40

avec un programme de canal simple et unique en utilisant un dispositif spécial de continuité de piste.

En plus des "composantes" de gestion des segments, de la mémoire principale et de la mémoire auxiliaire, le complexe de gestion de mémoire
5 virtuel comporte deux fonctions annexes. Le traitement des segments manquants, la gestion des piles.

5.1-4 - Traitement des segments manquants.

Comme précédemment mentionné, les descripteurs de segments contiennent tous une zone appelée bit de "présence" (P). Toute tentative effectuée en vue d'avoir accès à un segment de la mémoire principale qui a une entrée de descripteur $P = 0$ provoque la production d'une exception (appelée exception de segment manquant). Cette exception peut indiquer l'une des deux conditions : le segment n'est pas dans la mémoire principale ou il est provisoirement instable de telle sorte que l'accès audit segment doit être différé.

15 Il est fait appel à la composante de traitement des segments manquants du complexe de gestion de mémoire virtuelle lorsqu'une exception de segment manquant se produit et cette composante détermine celle des deux conditions qui est indiquée et prend la disposition appropriée pour la condition indiquée. Si le segment désiré n'est pas dans la mémoire principale, il est fait appel à la
20 fonction de "chargement du segment" (SGLD), qui sera décrite en détail, de la composante de gestion des segments. Si le segment est instable, le processus qui requiert l'accès au segment est placé en état d'attente jusqu'à ce que l'accès au segment soit à nouveau permis.

5.1-5 - Gestion des piles.

25 Pour accroître le rendement, des segments spécialisés de la mémoire principale appelés segments de pile peuvent être utilisés pour permettre de réaliser un mécanisme d'appel des procédures et pour fournir l'espace de travail pour les résultats intermédiaires de chaque procédure. Un tel segment de piles nécessite un support spécial qui n'est pas nécessaire pour les
30 autres segments et une composante de gestion de piles spéciale peut être utilisée pour remplir les fonctions requises pour la gestion de tels segments. En particulier, une composante de gestion de piles peut être utilisée pour détecter les conditions de débordement de pile et pour fournir un espace de mémoire supplémentaire dans lequel l'information débordant de la pile peut
35 être placée. Une description détaillée de telles procédures spécialisées n'est pas nécessaire pour la compréhension de la présente invention et ne sera donc pas donnée ici.

5.1-6 - Services de la mémoire virtuelle.

La dernière composante du complexe d'ensemble de gestion de mémoire virtuelle à laquelle on se référera peut être appelée "Services de la mémoire"
40

re virtuelle". Les services de la mémoire virtuelle contiennent une procédure simple unique appelée "extrait de mémoire virtuelle" (VM EXTR) dont la fonction est de mettre à la disposition des processus du système extérieurs au complexe de gestion de mémoire virtuelle les données disponibles dans la table de communications (COM-TAB, décrite au paragraphe 5.4.2. ci-après).
5 Par exemple, des procédures d'ordonnancement peuvent nécessiter la connaissance de l'information concernant l'importance de l'espace d'adresses disponible dans la mémoire principale. En utilisant la procédure d'extrait de la mémoire virtuelle et en spécifiant le mot clé UMAVLSZ (volume disponible de la mémoire des utilisateurs) l'information désirée est extraite de la
10 table des communications. Comme dans le cas de la gestion des piles, une explication détaillée des procédures des services de la mémoire virtuelle est inutile ici et ne sera donc pas donnée.

5.2 - Structure des procédures : Complexes, composantes primitives.

15 Un système d'exploitation global peut être divisé en zones, les zones en sections, les sections en complexes, les complexes en composantes, les composantes en primitives et les primitives en instructions exécutables. Par exemple, les procédures utilisées dans la gestion de la mémoire virtuelle comportant notamment la création, l'attribution, le partage et la translation
20 des segments peuvent être traitées par un unique "complexe" qui fait partie de la "section" machine étendue de la "zone" du dispositif de commande du "système" d'exploitation. Le complexe de gestion de mémoire virtuelle est, à son tour, divisé dans les six composantes décrites d'une manière générale ci-dessus : (1) gestion des segments, (2) gestion de la mémoire principale,
25 (3) gestion de la mémoire auxiliaire, (4) traitement des segments manquants, (5) gestion des piles et (6) services de la mémoire virtuelle.

Les composantes individuelles, telles que la gestion de la mémoire principale, constituent une série de procédures liées fonctionnellement et appelées ici primitives auxquelles des noms mnémoniques sont donnés. On
30 peut faire appel aux services fournis par une primitive en émettant une macro-instruction (ou en abréviation macro). Les macroinstructions établissent ainsi les jonctions entre les primitives à l'intérieur d'une composante et entre les composantes à l'intérieur d'un complexe. Certaines macroinstructions sont disponibles pour tous les utilisateurs tandis que des numéros
35 d'anneaux plus privilégiés sont attribués à d'autres qui ne sont en conséquence disponibles que pour les "utilisateurs du système". D'autres macroinstructions sont invisibles pour tous les utilisateurs, c'est-à-dire qu'elles sont totalement internes au système d'exploitation. Une macro-instruction utilisée pour faire appel à une primitive comprend, outre son nom mné-
40 monique, certains paramètres nécessaires au cours de l'exécution du proces-

sus de la primitive. Ces paramètres sont transmis au moyen de registres généraux (qui font partie des registres de commande de processus 120, Fig. 2).

Par exemple, comme on le décrira plus en détail ultérieurement, la composante de gestion de mémoire principale comporte une primitive appelée "allocation de mémoire utilisateur" identifiée par la mnémonique AUM. C'est la fonction de la primitive AUM d'attribuer un espace dans la mémoire principale à un segment qui doit être chargé. Les paramètres requis par la primitive AUM sont : (1) la dimension du segment qui doit être écrit dans la mémoire principale et (2) l'identité du groupe de processus (numéro J) pour lequel l'attribution doit être effectuée. Tous ces paramètres d'entrée sont mis à la disposition de la primitive du fait qu'ils ont été préalablement placés dans un registre général prédéterminé.

L'exécution de la primitive AUM donne l'adresse absolue de la zone de la mémoire principale attribuée au segment et le paramètre de sortie est, à son tour, mis en mémoire dans encore un autre registre général. S'il est désiré de retirer l'attribution de mémoire principale précédemment attribuée à un segment donné, la primitive de libération de la mémoire principale (RMM) est exécutée, cette primitive utilise comme paramètre d'entrée, l'adresse absolue du segment correspond précédemment mise en mémoire dans un registre général en tant que paramètre de sortie de la primitive AUM (le format utilisé pour la mise en mémoire des paramètres dans les registres généraux est décrit plus en détail au paragraphe 5.4.1 ci-après).

Dans le complexe de gestion de mémoire virtuelle spécifique mettant en oeuvre la présente invention, le complexe est découpé en six composantes, comportant chacune les primitives ayant les mnémoniques d'identification et noms énumérés ci-dessous :

Gestion de la mémoire principale

AUM	allocation de mémoire utilisateur
CUM	changement d'attribution de mémoire des utilisateurs
ACM	attribution de mémoire de commande
RMM	libération de mémoire principale

Gestion des segments

SGLD	chargement de segment
SGUNLD	déchargement de segment
SGSIZE	changement de dimension du segment
SGATTR	modification des attributs du segment
SDAVL	mise du bit de disponibilité
SDUNAVL	restauration du bit de disponibilité
RELVM	libération de la mémoire virtuelle par un groupe de processus

	ISDCRT	création de descripteur de segment indirect
	ISDDEL	suppression du descripteur de segment indirect
	ISDMOD	modification du descripteur de segment indirect
	SGMOVE	mouvement de segment
5	LDUNLD	déchargement segment spécial
	VMMBUF	contrôle des attributs tampon d'entrée/sortie
	SLDI	chargement de segment (interne)
	SUNLDI	déchargement de segment (interne)
	SSIZEI	changements des dimensions de segment (interne)
10	<u>Gestion de la mémoire auxiliaire</u>	
	ABSM	attribution de mémoire auxiliaire
	RBSM	libération de mémoire auxiliaire
	MVMAIN	mouvement segment à mémoire principale
	MVBACK	mouvement segment à mémoire auxiliaire
15	<u>Exception de segment manquant</u>	
	MISSGHDL	traitement de segment manquant
	<u>Gestion des piles</u>	
	STKOVHDL	traitement de débordement de pile
	RPSP	reprise d'un processus sur une procédure système
20	Le fonctionnement du complexe de gestion de mémoire virtuelle est pour la plus grande partie, invisible pour l'utilisateur. L'utilisateur ne dispose d'aucun moyen de connaître contrôler ou commander la manière suivant laquelle les segments qu'il crée sont échangés entre la mémoire principale et la mémoire auxiliaire ou de connaître leur organisation dans la mémoire	
25	principale physique. Il existe, cependant un nombre limité d'interfaces visibles pour l'utilisateur avec le complexe de gestion de mémoire virtuelle, les fonctions visibles à l'utilisateur permettant à l'utilisateur d'exécuter des instructions servant à charger des segments dans la mémoire principale, de désigner des segments comme étant éligibles pour être échangés et sortis de la	
30	mémoire principale et de changer la dimension et les attributs de segments désignés.	

5.3 - Interface entre les structures des procédures

Les diverses interfaces entre les composantes du complexe de gestion de mémoire virtuelle sont représentées dans le schéma-bloc de la fig.6.

- 35 Comme précédemment indiqué, les trois composantes principales du complexe de gestion de mémoire sont la composante de gestion des segments 251, la composante de gestion de la mémoire principale 252 et la composante de gestion de la mémoire auxiliaire 253. Chacune de ces trois composantes interroge et met à jour les structures d'informations y compris la base des
- 40 données des segments 254A comprise dans la composante 254 appelée

"services de la mémoire virtuelle".

En plus de l'interface visible pour l'utilisateur (mentionnée ci-dessus), le programme de l'utilisateur 250 peut faire appel aux procédures de gestion de mémoire virtuelle par le moyen du traitement des exceptions par l'inter-
5 médiaire de la composante de gestion des piles 255 et de la composante de segment 256.

Les primitives font appel aux services fournis par d'autres primitives dans la même composante ou dans des composantes différentes. Par exemple, la primitive SGLD (chargement segment) dans la composante de gestion des
10 segments fait appel à la primitive MVMAIN (mouvement segment à mémoire principale) qui se trouve dans la composante de gestion de la mémoire auxiliaire. Ainsi, la primitive SGLD est une primitive demanderesse et la primitive MVMAIN est une primitive "appelée". La matrice complète des primitives demanderesse et appelées dans le complexe de gestion virtuelle est représen-
15 tée sur la Fig. 7A.

Ainsi on peut constater que la composante de gestion des segments utilise des primitives dans les composantes de gestion de la mémoire principale et de gestion de la mémoire auxiliaire en plus des primitives qu'elle contient. Les composantes de gestion de la mémoire principale, de traitement des seg-
20 ments manquants et de gestion des piles font appel à des primitives contenues dans la composante de gestion des segments. La composante de gestion des piles fait appel à des primitives contenues dans la composante de gestion de la mémoire principale.

Les interfaces énumérées sur la Fig. 7A sont représentées graphique-
25 ment sur la Fig. 7B.

5.4 - Structure de données

Au cours de l'exécution des primitives contenues dans le complexe de gestion de mémoire virtuelle, il est nécessaire d'avoir accès aux données ci-après et de la modifier.

- 30 (1) paramètres mis dans les registres généraux.
(2) la table de communication (COM TAB).
(3) les en-tête des segments et des trous.
(4) les descripteurs de segments.
(5) les segments de commande de groupes de processus (PGCS).
35 (6) la table d'activité des travaux (JOT).
(7) la base du système du matériel; et
(8) les sémaphores du système (SYS1, SYSALL, SYSTACK1, SYSMMSAT et SYSBSIO).

5.4-1 - Registres généraux

40 Les registres généraux utilisés au cours du transfert des paramètres

d'une primitive à une autre sont identifiés par les noms mnémoniques et contiennent les informations ci-après :

- (1) GRJPS - contient l'identification des groupes de processus, des processus et des segments (J, P, STN, STE, MBZ).
- 5 (2) GRRC - contient un code de retour de format standard.
- (3) GRAA - contient l'adresse absolue des segments ou des descripteurs de segments.
- (4) GRCPAA - contient l'adresse absolue des zones de programmes de canaux attribuées.
- 10 (5) GRSZ - contient une dimension de segment.
- (6) GRMSZ - contient une dimension maximale de segment.
- (7) GRDSZ - contient le nombre de blocs ayant une longueur de 16 multipliets (octets) qui doivent être écrits sur la première piste de la mémoire auxiliaire, le nombre total de pistes nécessaires pour contenir la totalité du segment qui doit être écrit et le nombre de blocs de 16 multipliets qui doivent être écrits sur la dernière piste dans la mémoire auxiliaire.
- 15 (8) GRDA - contient l'adresse relative sur les disques d'un segment qui doit être accédé dans la mémoire auxiliaire exprimée sous forme de la combinaison du numéro du cylindre par rapport au début du fichier de la mémoire auxiliaire, du numéro de la tête sur le cylindre et du numéro d'enregistrement sur la piste où commence le segment (C, H, A).
- 20 (9) GRRO - contient l'adresse du disque (C, H, R) du bloc attribuable disponible suivant dans la mémoire auxiliaire.
- (10) GRTY - contient le type de demande et contient en outre des bits
- 25 Indicateurs indiquant les points ci-après :
 - a) DEBORDEMENT = 1 lorsque plus de cinq pistes sont nécessaires pour le segment à mettre en mémoire;
 - b) ECRIRE RO = 1 lorsqu'il est besoin de la commande de canal (CCE) écrire RO, ECRIRE RO = 0 lorsqu'il est besoin de la commande de canal (CCE)
 - 30 "LIRE RO".
 - c) MISE EN FORME = 1 pour les opérations de mise en forme.
 - d) BLOC = 1 si les dimensions du segment sont supérieures à celles du bloc attribuable.
 - e) MVMAIN/MVBACK = 1 pour les opérations de mouvement à la mémoire principale (MVMAIN) et = 0 pour les opérations de mouvement à la mémoire auxiliaire (MVBACK).
 - 35 f) MISE A JOUR DYNAMIQUE = 1 pour les opérations de mise à jour dynamique.
 - g) GESTION DES DONNEES = 1 pour les demandes de gestion des
 - 40 données.

h) CONSERVER = 1 si l'espace ne doit pas être libéré de la mémoire principale.

i) ACTION = 1 pour les opérations de lecture et = 0 pour les opérations de mise en place.

5 j) OBLIGATOIRE = 1 si la demande est obligatoire, 0 si la demande est informative.

k) CONTOURNEMENT = 1 lorsque le contournement doit être effectif.

l) IDENTIFICATION DE SEGMENT = 1 lorsque le type d'identification de segment est dans le format (STN, STE) = 0 lorsque le format d'identification de segment a la forme (J, PSTN, STE) et enfin,

10 m) SPECIAL = 1 pour des points d'entrée spécifiques dans la gestion de mémoire virtuelle.

Il n'est utilisé qu'un sous-ensemble de ces indicateurs dans les primitives individuelles de la gestion de mémoire virtuelle.

15 5.4 - 2 Table des communications

La table des communications (COM TAB) est utilisée collectivement par tous les processus et comporte les informations concernant l'état de la mémoire virtuelle. Les éléments de données numériques individuels contenus dans la table des communications sont énumérés ci-après avec leurs noms mnémoniques. La première liste identifie les éléments décrivant l'état de la mémoire principale tandis que la seconde liste identifie les éléments de données décrivant l'état de la mémoire auxiliaire.

Données de la table de communications (COM TAB) Partie concernant la mémoire principale

25	<u>Mnémonique</u>	<u>Description</u>
	CMAVLSZ	Dimension disponible de la mémoire de commande.
	CMBHSZ	dimension du plus grand trou de la mémoire de commande.
	CMFHADR	adresse du premier trou de la mémoire de commande.
30	CMLHADR	adresse du dernier trou de la mémoire de commande.
	UMAVLSZ	dimension disponible de la mémoire des utilisateurs.
35	UMBHSZ	dimension du plus grand trou de la mémoire des utilisateurs.
	UMFHADR	adresse du premier trou de la mémoire des utilisateurs.
40	UMBPHSZ	dimensions du plus grand pseudo-trou de la mémoire des utilisateurs.

	UMBPHADR	adresse du plus grand ^{pseudo} trou de la mémoire des utilisateurs.
	FWSZ	dimension de la programmation.
	NUSZ	dimension du noyau
5	CMADR	adresse de la mémoire de commande.
	UMISLADR	adresse d'origine de la mémoire des utilisateurs.
	UMCURADR	adresse courante de la mémoire des utilisateurs.
	FSGADR	adresse du premier segment de la liste des segments non éligibles pour l'échange (NESWLIST)
10	LSGADR	adresse du dernier segment de la liste NESWLIST.
	FESWADR	premier segment éligible pour être échangé dans la liste des segments éligibles pour être échangés (ESWLIST).
15	UMLAADR	adresse de la dernière zone de la mémoire des utilisateurs.
	MMEADR	adresse finale de la mémoire principale.
	MAXMAXSZ	dimensions la plus grande possible des segments.
	MMSZLL	limite inférieure du segment ayant la plus grande dimension possible.
20	MMSZLL	limite inférieure du segment ayant la plus grande dimension possible.
	MMSZUL	limite supérieure du segment ayant la plus grande dimension possible.
25	ROSTSZ	dimension du segment de pile d'anneau zéro.
	STPAGSZ	dimension de page de pile.
	CPAREAMP	plan de la zone des programmes de canaux.
	BSIOCNT	compteur entrée/sortie de la mémoire auxiliaire.
	BSIOMAX	valeur maximale d'opérations Entrée/Sortie de la mémoire auxiliaire.
30	MMWPRCT	nombre de processus attendant que la mémoire principale soit prête.
	THP	période de détection de l'emballement.
	THMAX	seuil maximal d'emballement.
	THMIN	seuil minimal de l'emballement.
35	THCNT	compteur de l'emballement.
	SWITCH 1	zone d'aiguillage 1.

Données de la table de communication (COMTAB) Partie concernant la mémoire auxiliaire.

	<u>Mnémonique</u>	<u>Description</u>
40	BS1ADR	partie 1 de la mémoire auxiliaire- adresse

		absolue du disque (MMCC).
	BS2ADR	partie 2 de la mémoire auxiliaire - adresse absolue du disque (MMCC).
5	BS3ADR	partie 3 de la mémoire auxiliaire - adresse absolue du disque (MMCC).
	BS2NAVLU	partie 2 de la mémoire auxiliaire - nombre des blocs attribuables disponibles.
	BSDUAS	mise à jour dynamique de la mémoire auxiliaire - état des attributions.
10	BSDUADR	mise à jour dynamique de la mémoire auxiliaire - adresse relative du disque (C, H, R).
	BS2NEXTH	partie 2 de la mémoire auxiliaire - tête suivante.
	BS2NU	partie 2 de la mémoire auxiliaire - nombre de blocs.
	BS3NU	partie 3 de la mémoire auxiliaire - nombre de blocs.
15	BS3NAVLU	partie 3 de la mémoire auxiliaire - nombre de blocs disponibles.
	BS3MAPCT	partie 3 de la mémoire auxiliaire - compteur de plan.
	BSPGCSSZ	dimensions de la sous-zone de mémoire auxiliaire dans le segment de commande des groupes de processus (PGCS) (mots).
20	BSNUPC	nombre de blocs/cylindres
	BSNTPU	nombre de pistes/blocs.
	BSNTPC	nombre de pistes/cylindres.
	BSNBPU	nombre de multiplats de la mémoire auxiliaire/blocs.
25	BSNBPT	nombre de multiplats de la mémoire ^{auxiliaire} /piste.
	BSMINU	nombre de blocs minimal de la mémoire auxiliaire.
	BSMAXU	nombre de blocs maximal de la mémoire auxiliaire.
	SWITCH 2	zone d'aiguillage 2.
	CCESK	squelette de l'entrée de commande de canal (CCE).
30	BS3MAPU	partie 3 de la mémoire auxiliaire - implantation des blocs.

Les dix neuf premières valeurs énumérées ci-dessus de la partie de mémoire principale de la table de communications (COM TAB) sont utilisées essentiellement par la primitive AUM (attribution de la mémoire des utilisateurs) de la gestion de la mémoire principale. La signification spécifique de chacune de ces 19 valeurs est représentée sur la fig. 8 qui représente l'état de la mémoire principale dans des conditions de fonctionnement typiques. Commenant à l'adresse physique 0 (au sommet de la fig. 8), la zone de matériel de traitement 201 s'étend jusqu'à l'adresse limite (BAR) et les dimensions de cette zone réservée au matériel FWSZ est mise en

mémoire dans la table COM TAB.

Le noyau du système 270 qui comporte la base du système 202, l'extension de la base du système 203 et la zone de mémoire de système d'exploitation résidant 204 (fig. 3), s'étend de la limite BAR jusqu'à l'adresse de la mémoire de commande CMADR et sa dimension est donnée par la dimension du noyau NUSZ.

La mémoire de commande 272 s'étend de l'adresse CMADR (adresse de la mémoire de commande) à l'adresse en cours de la mémoire des utilisateurs UMCURADR qui forme une limite mobile entre la mémoire des utilisateurs et la mémoire de commande.

La mémoire des utilisateurs 274 qui contient les segments échangeables s'étend de l'adresse UMCURADR à l'adresse de fin de la mémoire principale MMEADR.

Les régions de la mémoire principale dans lesquelles des segments ont été écrits sont représentés sur la Fig. 8 par des rectangles hachurés. Les segments écrits dans la mémoire des utilisateurs échangeable ont en outre été désignés comme étant soit éligibles pour l'échange (ESW) soit non éligibles pour l'échange (MESW). Les segments sont tout d'abord écrits dans la mémoire échangeable en commençant à l'adresse de fin de la mémoire principale (MMEADR) (au bas de la fig. 8). Ainsi, l'adresse du premier segment échangeable ESW, FESWADR est l'adresse du premier segment ESW à partir du bas de la Fig. 8.

5.4 - 3- Entêtes de segments et de trous.

Chaque segment écrit dans la mémoire principale qu'il contienne des données ou des procédures est précédé d'une entête de 32 octets composé de 8 mots de 4 octets W1-W8. Le format d'une entête de segment est représenté sur la fig. 9 et une description de la signification de chaque entrée de l'en-tête est donnée dans la liste ci-après.

Liste des données de l'en-tête de segment.

Mot 1: adresse absolue de la zone précédente (adresse de l'en-tête de la zone de la mémoire des utilisateurs / mémoire de commande suivante par ordre d'adresses décroissantes - remplie de 1 pour la première zone).

Mot 2: adresse absolue de la zone suivante (adresse de l'en-tête de la zone de la mémoire des utilisateurs / mémoire de commande suivante par ordre d'adresses croissantes - remplie de 1 pour la dernière zone).

Mot 3: adresse absolue du segment précédent (adresse de l'en-tête du segment de mémoire des utilisateurs suivant soit dans la liste des segments non éligibles pour l'échange NESWLIST (suivant la règle premier entré- premier sorti), soit dans la liste ESWLIST (segments éligibles pour l'échange) (par ordre d'adresses croissantes) - rempli de 1 pour le

premier segment de la liste.

Mot 4 : adresse absolue du segment suivant (adresse de l'en-tête du segment de mémoire des utilisateurs suivant soit dans la liste NESWLST (suivant la règle premier entré- premier sorti) soit dans la liste ESWLIST (par ordre d'adresses décroissantes) - rempli de 1 pour le dernier segment de la liste.

Mot 5 : bits 0 à 7 SGNALU : nombres de blocs attribués au segment dans le fichier de la mémoire auxiliaire.

bit 8 à 13 : non utilisés (réservés).

10 bits 14, 15 RF : coefficient de résidence. Cette zone contient la valeur du coefficient de résidence qui peut être égal à 0, 1, 2, 3 (0 signifiant la plus forte valeur de résidence etc.). Plus le coefficient de résidence est élevé, plus la probabilité qu'un segment reste dans la mémoire principale est faible.

15 bit 16 EXPD : Les dimensions de segment en cause ont varié (soit accrues soit diminuées).

bit 17 CODE : attribut non utilisé dans la gestion de la mémoire.

bit 18 Données : attribut non utilisé dans la gestion de la mémoire.

bit 19 Lien : attribut non utilisé dans la gestion de la mémoire.

bit 20 non utilisé.

20 bit 21 EXP : le segment en cause peut être agrandi (c'est-à-dire que la dimension du segment peut être modifiée); ce bit est mis par la préparation du programme.

bits 22, 23 PART : niveau d'utilisation collective; contiennent le type de table de segments auquel le segment appartient (0, 2 ou 3).

25 bit 24 : CS : Structure de commande (c'est-à-dire segment de commande du groupe du processus PGCS, bloc de commande de processus PCB).

bit 25 RELOC : le segment en cause est translatable (c'est-à-dire qu'il peut être implanté en un emplacement quelconque sans difficulté).

30 bit 26 RES : le segment changé d'emplacement doit être résidant dans la mémoire principale (c'est-à-dire qu'il n'est pas échangeable).

bit 27 PILE : le segment en cause est un segment de pile.

bit 28 CONT : le segment en cause est situé dans la mémoire de commande (par exemple le segment de commande de groupe de processus PGCS, segment de pile d'anneau "0", etc).

35 bit 29 : BUF : le segment en cause est un segment de mémoire tampon ou contient une ou des mémoires tampon.

40 bit 30 : ESW : le segment en cause appartient à la liste ESWLIST (c'est-à-dire que le segment est éligible pour être échangé lorsque ce bit est égal à 1) (ce bit est mis à l'état 1 lorsqu'une macro-instruction informative de la primitive SGUNLD (déchargement de segment) est émise pour

ce segment).

bit 31 ARTYP - type de zone - ce bit est égal à 1 lorsque l'en-tête est une en-tête de segment ou l'en-tête d'une zone qui ^{est} en cours d'attribution et égal à 0 lorsque l'en-tête est une en-tête de trou.

5 Mot 6 C-H-R : ces zones contiennent l'adresse relative du disque dans le fichier de la mémoire auxiliaire de la copie du segment correspondante dans la mémoire auxiliaire (le cas échéant).

Mot 7 MSIZE (dimension maximale) : dimension maximale du segment. Le format est le même que pour la dimension du descripteur de segment direct (DSD) (MSIZE/16)-1.

10 Mot 8 S, J, P : Cette zone contient les noms de segment correspondants J.P.S.

Ensuite la définition d'un attribut "VERROUILLE" peut être ajoutée, constituée par la combinaison des attributs de résidence en mémoire et de non-translation (RES= 1, RELOC= 0).

Liste des données de l'en-tête de trou

Mot 1 : adresse absolue de la zone précédente (adresse d'en-tête de la zone de la mémoire des utilisateurs/mémoire de commande suivante par ordre d'adresses décroissantes - remplie de 1 pour la première zone).

20 Mot 2 : adresse absolue de la zone suivante (adresse de l'en-tête de la zone de la mémoire des utilisateurs/mémoire de commande suivante par ordre d'adresses croissantes - remplie de 1 pour la dernière zone).

Mots 3, 4 : réservés pour une utilisation future.

25 Mot 5 : bit 31 ARTYP : type de zone, ce bit est égal à 1 lorsque l'en-tête est une en-tête de segment et égal à 0 lorsque l'en-tête est une en-tête de trou.

Mot 6 : bits 8 à 31 dimensions du trou, en-tête exclue.

30 Mot 7 : adresse absolue du trou précédent (adresse de l'en-tête du trou suivant de la mémoire des utilisateurs/mémoire de commande soit dans la liste des trous de la mémoire des utilisateurs UMHLLIST, soit dans la liste des trous de la mémoire de commande CMHLLIST, par ordre d'adresses croissantes).

35 Mot 8 : adresse absolue du trou suivant (adresse de l'en-tête du trou suivant de la mémoire des utilisateurs/mémoire de commande soit dans la liste UMHLLIST soit dans la liste CMHLLIST, par ordre d'adresses décroissantes).

5.4-4 - Descripteurs de segments

Les descripteurs de segments directs (DSD) et indirects (ISD) situés tous deux dans les tables de segments contiennent des informations utilisées dans les procédures de gestion de mémoire virtuelle. Le format des descrip-

40

teurs de segment indirect est représenté sur la fig. 11 et celui des descripteurs de segment direct est représenté sur la fig. 12. Les descripteurs de segments directs et indirects contiennent tous deux un bit P et un bit I respectivement aux positions binaires 0 et 1. Lorsque P = 1, la présence du segment dans la mémoire principale est indiquée. Le descripteur indirect est caractérisé par la présence d'un bit I = 1 à la position binaire 1 qui indique que le descripteur du segment est indirect.

La zone A située à la position binaire 3 tant dans le descripteur indirect que le descripteur direct est une zone d'un bit indiquant si le segment est ou non "disponible". Si l'indicateur A est à l'état 1, le segment auquel le descripteur correspond peut être accédé tandis que si A = 0 le segment est "verrouillé" et une référence au descripteur de segment provoque une exception de segment verrouillé.

La zone U qui apparaît à la position binaire 4 du descripteur de segment direct représenté sur la fig. 12 indique si le segment a ou non été précédemment utilisé. Si l'indicateur U est à l'état 0 le segment n'a pas été accédé tandis que si l'indicateur U est à l'état 1, il indique que le segment a été précédemment extrait de la mémoire principale.

L'indicateur W situé à la position binaire 5 du descripteur de segment direct est utilisé pour indiquer si une écriture a été faite ou non dans le segment.

La zone G-S aux positions 6 et 7 du descripteur direct DSD est interrogée lorsque certaines instructions doivent être exécutées. Par exemple, une instruction d'entrée d'une procédure (ENT) nécessite que la zone G-S contienne un code prédéterminé, une instruction sur les sémaphores nécessite un code différent etc. Tout code G-S autre que celui spécifié provoque une exception de descripteur de segment invalide.

Les zones P, I, U et W sont utilisées pour spécifier le caractère des segments qui sont traités. La fig. 13 représente un tableau montrant la manière suivant laquelle ces zones prises en combinaison, sont interprétées.

En plus de la partie de 32 bits des descripteur de segments tels que représentés sur la fig. 11 et 12, chaque descripteur de segment contient une zone supplémentaire de 32 bits. Dans le descripteur de segment indirect, cette zone de 4 octets supplémentaires contient les zones d'accès ainsi que les zones de gestion de données et de messages. Dans les descripteurs de segments directs le mot supplémentaire contient les zones d'accès et une mesure de la dimension du segment.

Comme représenté sur la fig. 14, le mot 0 du descripteur de segment direct 275, pointe l'adresse physique du segment 277, après l'en-tête de segment de 32 octets 279, tandis que le mot 1 du descripteur direct 275

contient la dimension réelle du segment 277, c'est-à-dire la dimension occupée dans la mémoire par le segment, moins les 32 octets de l'en-tête 279.

Lorsqu'un segment est "manquant" (c'est-à-dire qu'il se trouve dans la mémoire auxiliaire) son descripteur de segment direct contient l'adresse relative du disque (C, H, R) du segment dans la mémoire auxiliaire à la place de l'adresse de segment recherché dans la mémoire principale représentée sur la fig. 12.

5.4 - 5 - Segments de commande de groupe de processus

Le complexe de gestion de mémoire virtuelle peut accéder à un certain nombre de zones dans les segments de commande de groupe de processus. Ces segments sont situés dans la zone de la mémoire de commande 205 (fig. 3) et comportent les blocs de commande de processus (PCB) et les segments d'extension de bloc de commande de processus (PCBE). Les éléments spécifiques mis en mémoire dans le segment de commande de processus utilisés par le complexe de gestion de mémoire virtuelle sont énumérés ci-après.

Contenu des segments de commande de groupe de processus PGCS

	<u>Mnémonique</u>	<u>Description</u>
20	PGBSCNALU	nombre de blocs de la mémoire auxiliaire attribués au groupe de processus - spécifiant le nombre de blocs de la mémoire auxiliaire attribués à chaque groupe de processus.
	PGBSAS	état d'attribution de la mémoire auxiliaire au groupe de processus décrivant l'état d'attribution du bloc qui est en cours de remplissage.
25	PGTAV	pistes disponibles pour le groupe de processus - donnant le nombre de pistes disponibles dans le bloc qui est en cours de remplissage.
	PGBAV	Multiplats disponibles pour le groupe de processus donnant le nombre de multiplats disponibles dans la première des pistes disponibles.
30	PGBSADR	adresse du groupe de processus dans la mémoire auxiliaire - donnant l'adresse relative sur le disque (C, H, R) de l'enregistrement suivant qui doit être écrit dans le bloc qui est en cours de remplissage.
35	PGMAPU	plan des blocs du groupe de processus - indiquant les blocs occupés par le groupe de processus en cause.
40	SSPW1	mots de pool de segments de pile situés dans l'extension du bloc de commande de processus

(PCBE) indiquant l'adresse (numéro de table de segment STN, entrée de table de segment STE) de la première entrée dans le pool de segments de pile et également le numéro d'entrée dans la table des segments de la dernière entrée du pool. Les mots SSPW1 décrivent les emplacements réservés dans certaines tables de segments pour des descripteurs de segments qui peuvent être utilisés comme pointeurs pour les segments de dépassement de capacité dans le cas d'une mise en pages de la pile.

5.4. -6 - Table d'activité des travaux JOT

La gestion de mémoire virtuelle fait appel également à certaines informations mises en mémoire dans la table d'activité des travaux (JOT). La table d'activité des travaux contient des entrées indiquant l'importance de mémoire de commande et de mémoire des utilisateurs spécifiée pour le groupe de processus par l'utilisateur ainsi que l'importance effective instantanée de la mémoire principale qui est utilisée par un groupe de processus donné. Plus spécifiquement, la table JOT contient les données ci-après utilisées par la gestion de mémoire virtuelle.

Données de la table JOT

<u>Mnémonique</u>	<u>Description</u>
DWS	ensemble de travail déclaré donnant le volume de mémoire des utilisateurs spécifié pour un groupe de processus par l'utilisateur.
IWS	ensemble de travail instantané donnant la valeur instantanée correspondante de l'espace d'adresses occupé par le groupe de processus dans la mémoire principale.
CMDWS	ensemble de travail de mémoire de commande déclaré donnant le volume de mémoire de commande spécifié pour un groupe de processus donné.
CMIWS	ensemble de travail instantané de la mémoire de commande donnant la valeur instantanée correspondante.

La table d'activité des travaux JOT contient également un indicateur qui indique si le groupe de processus est ou non "SOCIABLE". Comme on le décrira plus en détail en se référant à la phase de la primitive AUM (attribution de la mémoire des utilisateurs) au paragraphe 6.2.1.1 ci-après, les groupes de processus "insociables" se voient garantir un espace dans la

mémoire principale au moins égal à leur ensemble de travail déclaré (DWS).

5.4. - 7 - Base du système du matériel

La gestion de mémoire virtuelle utilise également deux éléments d'informations mis en mémoire dans la base du système, le mot de table J et le
5 mot de processus en cours (cf fig. 4).

5.4 - 8 - Sémaphores du système

Enfin le complexe de gestion de mémoire virtuelle utilise des sémaphores pour établir les communications entre certains processus asynchrones. Ces sémaphores sont les suivants :

10 Sémaphores

Mnémoniques

Description

SYS1

un sémaphore sans message pour traiter la base de données de segments sur une base d'exclusion mutuelle.

15 SYSALL

un sémaphore sans message utilisé pour assurer une exclusion mutuelle lors de l'attribution ou du retrait d'attribution de la mémoire principale aux segments.

SYSMMSAT

20

un sémaphore sans message utilisé pour traiter la saturation de la mémoire des utilisateurs.

SYSBIO

un second sémaphore sans message utilisé pour traiter la saturation de la mémoire des utilisateurs.

SYSTACK1

25

un sémaphore avec message utilisé par la gestion de pile pour effectuer la jonction avec certaines routines du matériel lorsqu'une condition de débordement de pile est détectée.

5.4 - 9 - Résumé : Structure des données

Comme précédemment décrit, les primitives qui forment le complexe
30 de gestion de mémoire virtuelle ont accès à l'information contenue dans la table des communications (COM TAB), aux en-têtes des segments et des trous dans la mémoire échangeable, aux segments de commande de groupe de processus dans la mémoire de commande, aux descripteurs de segments directs et indirects contenus dans les tables de segments, à certains sémaphores (contenus
35 dans les segments G) et à deux mots de la base du système. La manière spécifique suivant laquelle chaque primitive utilise ces structures de données sera examiné au cours de la description détaillée de chaque primitive qui va suivre.

6 - Description dynamique des primitives

40

6 - 1 Résumé

Dans la description ci-après des primitives individuelles qui constituent les éléments du complexe de gestion de mémoire virtuelle, l'ensemble des règles de base sera tout d'abord expliqué dans ses grandes lignes d'une manière générale et il est possible d'obtenir une compréhension convenable de la présente invention à la seule lecture de ces grandes lignes. La description sera poursuivie cependant par une description détaillée d'un exemple de procédure travaillant dans le contexte du matériel de traitement et des structures de données spécifiques qui ont été décrits.

6 - 2 Gestion de la mémoire principale

6 -2 -1 - Attribution de la mémoire des utilisateurs (AUM)

6-2-1-1 - Description générale

Le but de cette procédure est de donner de l'espace, sur requête, dans cette partie de la mémoire principale (appelée mémoire des utilisateurs) qui est soumise à l'échange des segments.

Les segments sont entrés dans la mémoire des utilisateurs dans deux cas; ou bien une macro-instruction explicite de chargement de segment (SGLD) est émise par le processus de l'utilisateur (dans l'anticipation que ce segment sera nécessaire) ou une exception de segment manquant implicite est déclenchée au moment d'une tentative de référence à un segment précédemment retourné dans la mémoire auxiliaire. Dans ce dernier cas, les segments sont chargés automatiquement sur une base "à la demande" invisible à l'utilisateur.

Les segments peuvent être translatés dans la mémoire des utilisateurs afin d'améliorer le rendement de l'organisation de la mémoire. En outre les segments peuvent être échangés et sortis de la mémoire principale et ramenés dans la mémoire auxiliaire selon une politique de remplacement (que l'on décrira ultérieurement) qui spécifie quel est le segment ou quels sont les segments qui doivent être retirés de la mémoire principale afin de laisser la place aux segments qui sont nécessaires au processus en cours d'exécution.

Bien qu'on ait indiqué ci-dessus que les segments de la mémoire des utilisateurs étaient translatables et échangeables, une exception est faite en ce qui concerne les zones qui sont impliqués dans des opérations d'Entrée/Sortie entre la mémoire des utilisateurs et la mémoire auxiliaire. Chaque fois qu'un transfert de segment est entrepris (à partir de la mémoire auxiliaire ou vers la mémoire auxiliaire), la partie en cause de la mémoire des utilisateurs doit être verrouillée jusqu'à l'achèvement de l'exécution du programme du canal. Il en résulte que la compression complète de la mémoire par une translation des segments n'est effectuée que lorsque les opérations E/S en instance avec la mémoire auxiliaire ont été achevées.

Pour cette raison, la compression de la mémoire n'est effectuée, si elle l'est, que lorsque tous les autres moyens pour trouver un espace de mémoire suffisamment grand pour loger un nouveau segment ont échoué.

Un organigramme représentant les caractéristiques générales de la primitive AUM est représenté sur la fig. 16, tandis que les fig. 15A et 15E représentent la disposition de la mémoire des utilisateurs et les divers liens créés entre les segments et les trous par les informations contenues dans les en-têtes (Fig. 9, 10) et dans la table des communications qui servent de base à l'exécution de la primitive AUM.

La fonction d'attribution de la mémoire des utilisateurs peut être divisée en six phases successives.

Première phase Recherche d'un trou convenable.

La dimension du plus grand trou (espace d'adresses physiques vide) dans la mémoire des utilisateurs (UMBHSZ) est enregistrée dans la table des communications (COM TAB) (cf 5-4-2 ci-dessus). Cette dimension est comparée à la dimension demandée (DD) et s'il ^{en} résulte qu'il existe un trou suffisamment grand, la liste des trous est explorée en partant de l'adresse du premier trou de la mémoire des utilisateurs (UMFHADR) et en continuant par ordre d'adresses décroissantes (en utilisant les indicateurs qui se trouvent dans le mot 2 de chaque en-tête de trou - cf 5-4-3 ci-dessus) pour trouver un trou égal ou supérieur à la dimension demandée (DD). Pour cette raison les trous qui se trouvent le plus près de la fin (MMEADR) du sommet de la mémoire des utilisateurs sont attribués les premiers.

Si la dimension du trou le plus grand telle que donnée par la table COM TAB est plus petite que la dimension demandée (DD) la phase 2 est exécutée.

2) Seconde phase-règlage de la limite entre la mémoire de commande et la mémoire des utilisateurs.

Alors que la mémoire des utilisateurs est remplie en partant des adresses les plus grandes et en allant par ordre décroissant, la mémoire de commande est remplie en partant des adresses les plus petites et en allant par ordre croissant. Ainsi les espaces situés entre les parties remplies de la mémoire des utilisateurs et celles de la mémoire de commande peuvent, en combinaison, former un espace suffisant pour loger le segment dont le chargement est demandé. (On notera ici que ceci est le seul cas où deux trous adjacents peuvent coexister du fait que lorsque l'espace occupé par le dernier segment de commande dans le sens des adresses croissantes, est libéré, l'adresse de limite en cours (mise en mémoire en tant que UMCURADR dans la table COM TAB) entre la mémoire des utilisateurs et la mémoire de commande n'est pas automatiquement mise à jour). Si cette adresse de limite en cours est

supérieure à une limite inférieure (UMISLADR) fixée au moment où le système a été chargé, l'adresse de limite peut être alors abaissée, auquel cas la dimension du trou agrandi à l'extrémité voisine de la mémoire de commande de la mémoire des utilisateurs est comparée à la dimension demandée (DD).

Si cette dimension de trou est encore plus petite que la dimension demandée la phase 3 est exécutée.

Troisième phase - Recherche d'un pseudo-trou convenable

Lorsqu'un segment (ESW) éligible pour être changé (cf Fig. 13) pour la distinction entre les segments ESW et NESW) est adjacent à un trou (sur l'un ou sur les deux côtés du segment ESW) l'espace combiné est appelé un pseudo-trou. La dimension du plus grand pseudo-trou dans la mémoire des utilisateurs est enregistrée et disponible dans la table des communications. Cette dimension est comparée à la dimension demandée et si le pseudo-trou est suffisamment grand, la liste des segments éligibles pour l'échange (ESW LIST) est explorée, en commençant à l'adresse la plus grande, pour trouver un pseudo-trou plus grand ou égal à la dimension demandée. C'est-à-dire que l'exploration est commencée à l'adresse du premier segment éligible pour l'échange FESWADR (Fig. 8) et se poursuit en utilisant les indicateurs compris dans le mot 1 de chaque en-tête de segment ESW. En conséquence, les pseudo-trous qui se trouvent les plus près de l'adresse de la fin de la mémoire des utilisateurs sont attribués les premiers.

Si même la dimension du plus grand pseudo-trou est plus petite que la dimension demandée, la phase 4 est exécutée.

Quatrième phase - Libération des segments de piles de débordement inutilisés

La phase 4 est facultative et n'est pas utilisée dans les systèmes dans lesquels chaque pile d'anneau ne comporte qu'un segment.

Les segments de pile de débordement ayant des adresses identifiées par les mots de pool de segments de pile (SSPW1) dans l'extension du bloc de commande de processus (PCBE) peuvent être utilisés pour fournir un espace supplémentaire à une pile.

L'espace attribué aux segments de piles de débordement n'est pas libéré lorsqu'une pile est remontée à la fin d'un groupe de processus. Au contraire, l'espace n'est réattribué que lorsqu'il est nécessaire à un autre utilisateur. Du fait que la fonction d'attribution prend beaucoup plus de temps que la fonction de retrait d'attribution (ou libération), on fait une économie de temps dans la procédure de gestion de piles (si l'exception du débordement de pile se produit à nouveau) sous réserve que l'espace occupé n'ait pas été réattribué entre temps.

Chaque fois qu'un segment de débordement est transformé en trou, les dimensions du plus grand trou et du plus grand pseudo-trou sont à nouveau comparées à la dimension demandée. S'il n'y a plus de segment de débordement libérable et que la demande n'est pas satisfaite, la phase 5 est exécutée.

Cinquième phase - Recherche d'un trou virtuel convenable.

La liste des segments NESW (non éligibles pour l'échange) est explorée dans l'ordre premier entré-premier sorti, en commençant à l'adresse FSGADR (adresse du premier segment) dans la table des communications (COM TAB) et en parcourant la liste jusqu'aux segments les plus récemment ajoutés en utilisant le mot 4 des en-têtes des segments NESW (non éligibles pour l'échange). La dimension du trou virtuel est calculée pour chaque segment NESW échangeable (en ajoutant la dimension du segment NESW à celle du pseudo-trou adjacent) et le meilleur "candidat" à échanger est trouvé.

Afin de trouver le meilleur candidat, on tient compte des paramètres ci-après: le coefficient de résidence, la dimension du trou virtuel et la dimension du segment NESW.

Lorsque deux segments NESW de longueur inégale sont candidats pour satisfaire la demande et à condition que tous les autres paramètres soient identiques (dimension du trou virtuel suffisante, même coefficient de résidence etc...) leur dimension est comparée à celle de la dimension demandée multipliée par deux. Si la dimension de l'un est inférieure à la dimension demandée multipliée par deux et que celle de l'autre est supérieure à cette valeur, le plus petit des segments est échangé. Si les segments sont tous deux inférieurs à cette valeur, le plus ancien candidat dans la liste premier entré-premier sorti est choisi pour l'échange.

En outre, dès qu'un candidat est trouvé dont le facteur de résidence est égal à 3 et qu'il a une dimension inférieure au double de la demande l'exploration de la liste premier entré/premier sorti est arrêtée.

Sinon, l'exploration de la liste premier entré-premier sorti continue jusqu'à la fin en vue de trouver le meilleur candidat.

Les règles principales qui gouvernent la politique de remplacement pour le choix entre les segments NESW peut être définie comme suit :

- a) le coefficient de résidence est prépondérant dans tous les cas;
- b) lorsque la liste premier entré-premier sorti est explorée, le meilleur candidat est celui qui a la plus petite dimension.

c) parmi plusieurs candidats ayant une dimension de segment inférieure ou égale au double de la dimension demandée et le même facteur de résidence, la règle premier entré - premier sorti s'applique.

Si un espace est trouvé qui satisfait la demande les en-têtes de

segments et de trous sont mises à jour et l'adresse absolue du futur segment est retournée au demandeur. Si la phase 5 ne fournit pas l'espace demandé, la phase 6 est exécutée.

Sixième phase-Echange général et compression de la mémoire

5 Cette sixième phase est exécutée si la demande d'espace de mémoire principale ne peut être satisfaite par l'échange d'un unique segment NESW.

Dans ces conditions, les opérations suivantes sont exécutées.

La mise en route des groupes de processus (PGI), un ensemble de procédures utilisées pour préparer un groupe de processus à son exécution, reçoit l'ordre d'arrêter tout lancement de groupe de processus.

Un indicateur est mis pour interdire tout lancement d'opérations E/S avec la mémoire auxiliaire (sauf celles nécessaires à cette phase en cours).

Le traitement est arrêté jusqu'à achèvement des opérations E/S en cours de la mémoire auxiliaire.

15 Les segments sont déchargés de la mémoire principale en partant des adresses supérieures jusqu'à ce que le compteur de "dimension de mémoire des utilisateurs disponibles" atteigne une valeur égale ou supérieure à la dimension demandée et la compression de la mémoire est alors effectuée.

20 6.2 - 1 - 2 - Détails du traitement (Attribution de la mémoire des utilisateurs).

6.2 - 1-2-1 - Phase 1

1) La commande exclusive du traitement d'attribution est prise par l'intermédiaire du sémaphore sans message appelé SYSALL. Ensuite, la commande exclusive de la base des données des segments est prise, à nouveau par l'intermédiaire du sémaphore SYS1 (ces deux mémoires de système sont toujours appelées dans le même ordre). Le sémaphore SYSALL est utile du fait que la commande du sémaphore SYS1 peut être libérée ultérieurement, lorsque la fonction de déchargement des segments est appelée. En conséquence, pendant que les opérations E/S avec la gestion de la mémoire auxiliaire sont en cours d'exécution, les fonctions de la primitive d'attribution de la mémoire des utilisateurs (AUM) peuvent se poursuivre.

2) Afin d'empêcher le "pompage des segments" (va et vient des segments), les segments du demandeur qui sont nécessaires au temps de SORTIE (par exemple ceux contenant le point de retour et le segment de pile en cours) sont rendus résidants dans la mémoire des utilisateurs (en mettant à l'état 1 le bit 26 du mot 5 de l'en-tête de segment, cf 5, 4, 3 ci-dessus) ces segments continuent d'être résidants jusqu'à l'achèvement de l'attribution de la mémoire AUM. Si l'un de ces segments est manquant (comme indiqué par le bit de présence P du descripteur direct DSD), il est tout d'abord chargé dans la mémoire principale puis rendu résidant.

3) La dimension du plus grand trou de la mémoire des utilisateurs (UMBHSZ dans la table COM TAB) est comparée à la dimension demandée (DD) pour déterminer si la liste des trous de la mémoire des utilisateurs doit être explorée.

5 4) L'adresse du premier trou dans la mémoire des utilisateurs (UMFHADR) est extraite de la table des communications et mise en mémoire dans une zone de travail (appelé SAVADR 2) et l'en-tête du premier trou est accédée et mise en réserve dans une zone de travail (appelée DIM).

10 5) La dimension demandée est comparée à la dimension (DIM) pour déterminer si le trou a une dimension suffisante.

6) Si le trou n'est pas suffisamment grand, l'adresse du trou suivant est extraite (mot 2 dans les en-têtes de trous).

15 7) Si la nouvelle en-tête de segment doit remplacer l'en-tête de trou, le trou est enlevé de la liste des trous et les liaisons avec les en-têtes des trous précédents et suivants sont mises à jour. Si le trou supprimé était le premier de la liste, l'entrée correspondante (UMFHADR) dans la table COM-TAB est mise à jour. L'ancienne en-tête de trou devient une en-tête de segment (ARTYP = 1). Afin d'éviter que l'algorithme d'échange tienne compte du nouveau segment l'attribut VERROUILLE (c'est-à-dire RES = 1, RELOC = 0; ou résidant non translatable cf 5, 4, 3) est placé dans l'en-tête.

20 8) Si la nouvelle en-tête de segment est distincte de l'en-tête de trou, la procédure se poursuit comme suit: la dimension du nouveau trou est calculée en diminuant la dimension antérieure de la longueur qui doit être occupée par le nouveau segment.

25 9) le squelette de la nouvelle en-tête de segment est construit, une nouvelle zone est créée et liée aux zones précédentes et suivantes. Le nouveau segment est cadré par rapport aux adresses supérieures. L'adresse de la dernière zone de la mémoire des utilisateurs est mise à jour si nécessaire et l'indicateur de type de zone "ARTYP" est mis à l'état 1 (cf 5, 4, 3, ci-dessus). L'attribut VERROUILLE (RES) est également mis à l'état 1 dans l'en-tête du segment.

30 10) Le numéro J du propriétaire (groupe de processus) du nouveau segment est mis en mémoire dans la zone J de l'en-tête de segment qui vient d'être créée. Par convention, si le segment est du type 0, la zone J est remplie de 1.

11) Le nouveau segment "vide" est ajouté à la fin de la liste NESW-LIST.

40 12) L'adresse absolue du segment (telle qu'elle se trouve dans le descripteur de segment direct DSD) est mise en mémoire dans le registre GRAA.

13) Si le trou antérieur était le plus grand, la nouvelle liste des trous est explorée en partant des adresses supérieures, la dimension du plus grand/^{trou}de la mémoire des utilisateurs est calculée et le résultat est mis en mémoire dans la table COM TAB pour mettre à jour l'information UMBHSZ.

5 14) Un test est effectué pour déterminer si le trou antérieur appartenait au plus grand pseudo-trou (l'adresse de l'en-tête du trou est comparée avec les valeurs de l'information UMBPHADR) (plus grand pseudo-trou de la mémoire des utilisateurs) et UMBPHADR + UMBHSZ).

10 15) Si le test ci-dessus est positif, la mémoire des utilisateurs est explorée en suivant la liste ESWLIST, en commençant par les adresses supérieures et le nouveau plus grand pseudo-trou est localisé. (On notera que si plusieurs pseudo-trous ont la même dimension et sont en conséquence candidats pour être le plus grand, l'adresse conservée dans la table (COM TAB) est celle du candidat pseudo-trou ayant la plus petite adresse.
15 On notera en outre que la dimension d'un pseudo-trou est celle de l'espace libre effectif diminué des 32 octets de l'en-tête).

16) Le volume de mémoire des utilisateurs est diminué de la dimension DIM ou DIM + 32 selon le résultat de l'essai effectué à l'étape 6 ci-dessus.

20 17) L'indicateur du système IOSTOP (disposé dans la zone d'aiguillage 2 de la table (COM TAB)) est contrôlé pour déterminer si la poursuite de l'activité avec le fichier de la mémoire auxiliaire doit être reprise (elle a pu être arrêtée par suite de la saturation de la mémoire).

25 18) Si le test précédent, est positif l'indicateur IOSTOP est remis à zéro et la mise en route des groupes de processus (PGI) est informée que les groupes de processus peuvent être remis en route (l'activité PGI avait, dans ce cas, été précédemment arrêtée par suite du manque de mémoire des utilisateurs disponibles).

30 19) Si des processus sont en attente sur le sémaphore SYSMMSAT (comme indiqué par l'information MMWPRCT (nombre de processus en attente pour la mémoire principale) dans la table (COM TAB) du fait que la saturation de la mémoire a été détectée et qu'ils ont reçu l'interdiction d'entreprendre de nouvelles opérations E/S avec la mémoire auxiliaire; ils sont alors remis en activité.

35 20) Si ce n'est pas le premier appel de la fonction d'attribution de mémoire des utilisateurs (AUM), un retour est effectué immédiatement et les étapes 21 et 22 sont sautées (car la demanderesse doit conserver la commande du sémaphore (SYSALL).

21) La commande exclusive de la fonction d'attribution de la mémoire des utilisateurs est libérée par une notification au sémaphore SYSALL.

40 22) Les segments du demandeur qui ont été rendus résidents à

l'étape 3 sont à nouveau rendus échangeables en remettant à l'état 0 le bit de résidence (position 26 du mot 5 de l'en-tête des segments).

6 - 2 - 1 - 2 Phase 2

23) Les adresses UMISLADR et UMCURADR sont extraites de la table
5 des communications.

24) L'adresse en cours (UMCURADR) est nécessairement supérieure ou égale à l'adresse UMISLADR fixée à la création du système. Si ces adresses sont égales la phase 3 est immédiatement exécutée.

25) La valeur d'adresse UMCURADR est mise en mémoire dans une
10 zone de réserve (SAVADR 1).

26) L'adresse de la dernière zone de la mémoire de commande est trouvée dans le premier mot de la première zone de la mémoire des utilisateurs désignée par UMCURADR. Cette adresse est mise en mémoire dans une zone appelée COMMENCEMENT.

27) Si la dernière zone de la mémoire de commande n'est pas un trou
15 la phase 3 est exécutée.

28) Le début du dernier trou de la mémoire de commande est comparé à l'adresse établie lors du chargement de la mémoire des utilisateurs (UMISLADR).

29) Si l'adresse COMMENCEMENT est supérieure ou égale à UMISLADR,
20 le dernier trou dans la mémoire de commande est au-delà de l'adresse fixée lors du chargement UMISLADR et en conséquence le trou peut être enlevé de la liste des trous de la mémoire de commande CMLLIST. L'adresse du premier trou de la mémoire de commande CMFMADR est mise à jour avec
25 l'adresse du trou suivant (et l'information sur la dimension du plus grand trou de la mémoire de commande CMBHSZ est mise à jour si nécessaire).

30) La dimension du premier trou de la mémoire de commande est comparée à 32 (dimension de l'en-tête) pour déterminer s'il peut contenir une en-tête.

31) L'adresse réelle du trou est comparée à l'adresse UMISLADR.

32) Le premier trou de la mémoire de commande est réduit à son en-tête (dimension DIM = 0).

33) Il est déterminé si un espace suffisamment existe dans le trou au-delà de l'adresse UMISLADR pour contenir au moins une nouvelle en-tête.

34) La dimension du premier trou de la mémoire de commande est
35 mise à jour.

35) La dimension du nouveau/dernier trou de la mémoire des utilisateurs est alors calculée comme suit :

$$\text{DIMENSION} = \text{UMISLADR} - \text{SAVADR 1} - 32$$

40 (dimension du trou = adresse d'origine de la mémoire des utilisateurs

moins adresse en cours de la mémoire des utilisateurs mise en réserve, moins en-tête).

36) Les volumes de mémoire disponibles sont modifiés par la quantité DIMENSION + 32.

5 37) La dimension du plus grand trou de la mémoire de commande est modifiée si nécessaire.

38) Une nouvelle en-tête de trou est créée au début de la mémoire des utilisateurs (UMCURADR). Une nouvelle zone est ainsi créée et liée aux autres.

10 39) Le type de la zone désignée par la valeur d'adresse mise en réserve SAVADR 1 est déterminé.

40) Les deux trous adjacents sont réunis. La nouvelle dimension du trou est la somme des deux trous précédents.

15 41) Le nouveau trou a une dimension égale à "DIM", il est lié au trou précédent dans la mémoire des utilisateurs.

42) La dimension du plus grand trou de la mémoire des utilisateurs est mise à jour^{si}/nécessaire.

43) La dimension et l'adresse du plus grand pseudo-trou de la mémoire des utilisateurs sont mises à jour si nécessaire.

20 44) La dimension du nouveau trou est comparée à la dimension demandée (DD). Si le trou est suffisamment grand, l'espace est attribué au demandeur.

45) L'adresse de l'en-tête du nouveau trou de la mémoire des utilisateurs est mise en mémoire dans la zone SAVADR 2.

25 6 - 2 - 1 - 2 - 3 - Phase 3

46) Il est déterminé si la liste ESWLIST est vide ou non.

47) Dans la négative, la dimension du plus grand pseudo-trou (UMBPHSZ) est comparée à la dimension demandée (DD); si elle est plus petite la phase⁴ est exécutée.

30 48) La liste ESWLIST est explorée en commençant à l'adresse FESWADR (dans la table COM TAB) et l'adresse du premier segment est mise en mémoire dans la zone SAVADR 1.

35 49) La dimension du segment ESW est calculée (adresse de la zone suivante - mot 2 de l'en-tête - adresse du segment) et le résultat est mis en mémoire dans la zone de travail DIMENSION.

50) L'adresse du segment suivant dans la liste ESWLIST est mise en mémoire dans la zone de travail SAVADR 2.

51) L'exploration de la liste ESWLIST est effectuée de telle sorte que seulement une zone adjacente est prise en considération à chaque fois.

40 Si cette zone est un trou, la valeur DIM mise en mémoire dans la zone de

travail est accrue de la dimension du trou.

52) Les zones dans l'ordre d'adresses décroissantes sont explorées afin de calculer la dimension actuelle du pseudo-trou. La dimension totale du pseudo-trou est calculée afin d'optimiser le temps d'exploration de la mémoire des utilisateurs.

53) La dimension du segment ESW est calculée et ajoutée à la valeur DIM et l'adresse du segment ESW suivant est mise en mémoire dans la zone de travail SAVADR 2.

54) La dimension calculée du pseudo-trou est comparée à la dimension demandée (DD) et si elle est trouvée trop petite, le segment ESW suivant est accédé et la zone de travail SAVADR1 est rechargée.

55) La primitive SGUNLD (déchargement de segment) de la composante de gestion des segments (cf 6, 3, 2 ci-après) est appelée pour échanger le segment ESW. Lorsque la fonction SGUNLD a été exécutée, ou bien l'en-tête du segment à décharger devient l'en-tête du nouveau trou, ou si le segment à décharger était précédé par un trou adjacent (comme déterminé par la valeur du bit ARTYP de l'en-tête de la zone précédente lue au moyen de l'adresse de la zone précédente trouvée de l'ancienne en-tête du segment ESW) l'adresse de la zone précédente et la nouvelle dimension du trou sont placées dans l'en-tête du nouveau trou.

56) La boucle y compris la primitive SGUNLD est reprise jusqu'à ce que la dimension du trou dépasse la dimension demandée (DD).

6 - 2 - 1 - 2 - 4 - Phase 4

Comme précédemment indiqué au paragraphe 6, 2, 1, 1, la phase 4 est facultative et n'est pas utilisée à moins que des segments de piles mis en pages soient utilisés. Si cette phase 4 n'est pas utilisée la fonction AUM passe à la phase 5 (6, 2, 1, 2, 5 ci-après).

57) La liste NESWLIST est explorée en commençant à partir du segment le plus ancien. Seuls les segments de piles (indiqués par le bit 27 mot 5 de l'en-tête) sont pris en considération et les segments résidants sont ignorés.

58) L'identification complète du segment est extraite de l'en-tête du segment de pile.

59) Si le propriétaire du segment est le processus en cours, le numéro de segment de pile en cours (STNT, STET) est extrait du registre T et la variable du numéro d'anneau en cours (R) est mise initialement à zéro.

60) Si le propriétaire du segment n'est pas le processus en cours, le numéro du segment de pile en cours est trouvé (dans le mot de pile du bloc de commande du processus PCB, cf 5, 4, 5 ci-dessus) et le numéro d'anneau en cours est extrait.

61) Le mot de base de pile (SBW1) et le mot de pool de segments de pile (SSPW1) correspondant à l'anneau R sont lus dans le segment de commande de processus du propriétaire du segment.

5 62) S'il existe un pool de l'anneau R (SSPW \neq 0), il est alors déterminé si le segment de pile en cours est un segment de pile de base.

63) L'entrée correspondante au segment qui est actuellement utilisé est sautée.

64) Un contrôle est effectué pour déterminer si toutes les entrées ont été prises en considération dans le pool en cause.

10 65) Les bits sont contrôlés dans la description du segment pour déterminer si le segment est présent, manquant ou est un segment fantôme (cf fig. 13).

15 66) Si les segments qui viennent d'être examinés appartiennent au pool d'anneau 0, l'exploration est terminée pour le processus propriétaire et les pools des anneaux plus privilégiés sont alors examinés.

67) L'espace occupé par le segment de pile de débordement est libéré (sans transfert avec la mémoire auxiliaire) et le descripteur du segment en cause devient un descripteur du segment fantôme (comme à l'état initial).

20 68) Il est déterminé si le plus grand pseudo-trou et trou ont maintenant une dimension suffisante. Dans la négative, le traitement de la phase 4 est repris avec l'élément suivant dans la liste des segments.

6 - 2 - 1 - 2 - 5 - Phase 5

69) Le premier segment de la liste NEWSLIST est extrait.

70) L'indicateur SW1 de la table COM TAB est mis à zéro.

25 71) Le bit "SOCIABLE" dans la table d'activité des travaux (JOT) (cf paragraphes 5, 4, 6 ci-dessus) du groupe de processus en cours est contrôlé.

30 72) Si l'ensemble de travail déclaré (DWS) dans la table JOT est supérieur ou égal à l'ensemble de travail instantané qui se trouve également dans la table JOT, dans ce cas le groupe de processus en cours ne doit avoir aucun de ses segments échangés.

35 73) Si l'ensemble de travail instantané (IWS) est plus grand que l'ensemble de travail déclaré (DWS), seuls les segments appartenant au groupe de processus en cours peuvent être échangés. Si le segment extrait de la liste NEWSLIST n'appartient pas au processus en cours, l'élément suivant de la liste NEWSLIST est extrait (saut à l'étape 80 ci-après).

74) Si l'ensemble IWS est supérieur à l'ensemble DWS et que le segment extrait de la liste NEWSLIST appartient au processus en cours ou si le processus en cours appartient à un groupe de processus SOCIABLE.

40 75) Si $P = 0$ ou $A = 0$ ou $U = 1$ (cf 5, 4, 4) le bit U d'utilisation

est remis à zéro s'il était à l'état 1, et le segment suivant de la liste NESWLIST est extrait (à l'étape 80); et

76) si $P = 1$ et $A = 1$ et $U = 0$ les indicateurs DEPLACABLE sont contrôlés.

5 77) Si le segment n'est pas déplaçable, le segment suivant de la liste NESWLIST est extrait (à l'étape 80).

78) Si le segment est DEPLACABLE, le bit SOCIABLE du groupe de processus propriétaire est vérifié.

10 79) Si le processus propriétaire appartient à un groupe de processus INSOCIABLE la dimension "Ensemble de travail instantanée (IWS) - dimension du segment extrait de la liste NESWLIST est calculée et comparée à l'ensemble de travail déclaré DWS. Si le résultat reste supérieur à DWS, le segment extrait de la liste NESWLIST peut être échangé. Si cependant le résultat est inférieur à DWS, le segment ne peut être échangé et le segment
15 suivant de la liste NESWLIST est extrait.

80) L'exploration se poursuit avec le segment suivant de la liste premier entré-premier sorti.

81) L'indicateur SW1 est vérifié pour déterminer si un trou virtuel satisfait la demande. Dans la négative la phase 6 est exécutée.

20 82) Si un trou virtuel a été choisi, les segments qui en font partie sont alors déchargés.

83) La dimension du trou virtuel en cause est calculée; si elle est inférieure à la dimension demandée, le segment suivant de la liste NESWLIST est lu.

25 On notera ici que l'indicateur SW1 peut avoir trois valeurs : "0" signifie: pas de trou virtuel choisi; "1" signifie : trou virtuel choisi comprenant un segment NESW d'une dimension supérieure au double de la dimension demandée et "2" signifie : trou virtuel choisi comprenant un segment NESW ayant une dimension inférieure ou égale au double de la dimension demandée.

30 On rappellera que tous les trous virtuels comportant un segment NESW ayant le même coefficient de résidence et une dimension inférieure à deux fois la dimension demandée sont équivalents pour la politique de remplacement.

Le coefficient de résidence RF et la dimension du segment (SSIZE) caractérisent le trou virtuel choisi. Le coefficient de résidence est le
35 facteur prépondérant pour la sélection du meilleur candidat (RF = 0 signifie hautement résidant RF = 3 est l'option par défaut du compilateur).

84) Lorsque le coefficient de résidence $RF = 0$ et que l'indicateur $SW1 = 2$ (c'est-à-dire que la dimension du segment (SSIZEA) est inférieure ou égale à 2 fois la dimension demandée) tous les trous virtuels
40 sont équivalents. Leur rang dans la liste premier entré- premier sorti

sert alors à choisir le segment le plus ancien.

85) La dimension du segment lu dans la liste NESWLIST (SSIZE) est comparée au double de la dimension demandée. Si elle est supérieure, l'indicateur SW1 conserve la valeur 1 et la dimension du segment est le facteur qui permet de choisir le meilleur candidat. Si la dimension SSIZE est plus petite ou égale à 2DD, l'indicateur SW1 prend la valeur 2.

86) Si le coefficient de résidence RF = 3 (c'est-à-dire la valeur la plus élevée) et du fait que l'indicateur SW1 = 2, le trou virtuel examiné devient le trou choisi et l'exploration de la liste des segments est arrêtée.

87) La dimension du segment NESW est comparée au double de la dimension demandée et l'indicateur SW1 est mis à l'état correspondant.

88) Les paramètres RF (coefficient de résidence) et SSIZE (dimension de segment) caractérisant le candidat choisi sont mis en mémoire dans deux variables RF0 et SSIZE0.

6 - 2 - 1 - 2 - 6 Phase 6

La phase six est exécutée lorsqu'il existe une saturation de la mémoire.

89) L'indicateur IOSTOP (arrêt Entrée/Sortie)^{de} la zone d'aiguillage SW2 est mise à l'état 1 pour interdire toute nouvelle mise en route d'opération E/S avec le fichier de la mémoire auxiliaire. (Cet indicateur est vérifié au cours des deux fonctions de chargement de segment (SGLD) et de déchargement de segment (SGUNLD) (cf 6, 3, 1 et 6, 3, 2 ci-après).

90) La mise en route des groupes de processus est notifiée pour qu'elle arrête tout lancement de groupe de processus du fait du manque de mémoire des utilisateurs.

91) Le compteur des opérations E/S en instance avec la mémoire auxiliaire (BSIOCNT) situé dans la table COM TAB est vérifié pour déterminer s'il existe encore des segments bloqués dans la mémoire des utilisateurs.

92) Le processus en cours est mis en attente sur le sémaphore SYSBS10 jusqu'à ce que toutes les opérations E/S en instance soient achevées.

93) L'adresse de la dernière zone est extraite de la table COM TAB.

94) Si la dimension de mémoire des utilisateurs disponible (la somme de tous les trous répartis dans la mémoire des utilisateurs) est supérieure ou égale à la dimension demandée, la compression de la mémoire est effectuée en partant des adresses supérieures pour libérer l'espace requis pour le segment désiré.

95) Dans la négative, la liste des zones est explorée en partant des adresses supérieures, les trous et les segments résidants étant ignorés

GRSZ - (FOLAR - GRAA)

3) Si l'accroissement est négatif, les informations ci-après sont mises à jour: liste des trous de la mémoire des utilisateurs (UMHLLIST) indicateur de la zone suivante (FOLAR) dans l'en-tête du segment Sx; dimension de la mémoire des utilisateurs disponible (UMAVLSZ): dimension du plus grand trou de la mémoire des utilisateurs (UMBHSZ), si nécessaire; adresse du premier trou de la mémoire des utilisateurs (UMFHADR), si nécessaire; dimension du plus grand pseudo-trou de la mémoire des utilisateurs (UMBHSZ), si nécessaire: et adresse de la dernière zone de la mémoire des utilisateurs (UMLAADR), si nécessaire.

4) Un code de retour (RCO) indiquant que le segment reste au même emplacement après son changement de dimension est envoyé au demandeur.

5) Si l'accroissement est positif, la commande exclusive de l'attribution de la mémoire des utilisateurs du sémaphore SYS1 est abandonnée et reprise après prise de commande exclusive du sémaphore SYSALL.

6) Un pointeur PTRS qui sera utilisé pour désigner le segment supérieur intéressé éligible pour être échangé dans le cas où la primitive AUM n'est pas appelée est mis en place avec l'adresse absolue du segment en cause Sx.

7) L'indicateur SW1 est mis à l'état 0 et l'ancienne dimension du segment (OLDSGSIZE) est transférée à la zone de travail "DIM".

8) Si la zone suivant le segment Sx est un trou, la dimension de l'espace disponible est mise à jour.

9) Si la nouvelle dimension DIM est inférieure à la dimension demandée, il n'y a pas suffisamment d'espace et une nouvelle phase de recherche est entreprise.

10) Le pointeur PTRS (dont la valeur peut avoir été changée à l'étape 17 ci-après) est comparé à la valeur du registre GRAA. Si le pointeur est égal à la valeur contenue dans le registre GRAA, il n'existe pas de segment à décharger pour satisfaire la demande. Si le pointeur PTRS n'est pas égal à la valeur de registre GRAA, on passe à l'étape 19 ci-après.

11) L'accroissement est comparé à la nouvelle dimension de trou et si l'accroissement est supérieure à la dimension de trou, il y a suffisamment d'espace pour accroître la dimension du segment mais pas assez pour établir une nouvelle en-tête de trou. Dans ce cas, tout l'espace est attribué au segment et aucun nouveau trou n'est créé.

L'ancien trou est enlevé de la liste UMHLLIST et cette liste est mise à jour. La liste des zones et la table des communications sont mises à jour, comme dans l'étape 3.

- 12) Si l'accroissement est inférieur ou égal à la dimension du trou, les pointeurs du trou sont mis à jour dans la liste UMHLLIST, la dimension du trou en cause est mise à jour (nouvelle dimension = dimension du trou - accroissement) et la liste des zones de la mémoire ainsi que la table des communications sont mises à jour, comme à l'étape 3 ci-dessus.
- 13) Le code de retour RC0 est chargé dans le registre GRRC.
- 14) Si la dimension DIM (étape 9) est inférieure à la dimension demandée et si l'indicateur SW1 = 0, cet indicateur est mis à l'état 1.
- 15) Si la dimension du plus grand trou de la mémoire des utilisateurs (UMBHSZ) est supérieure ou égale à la dimension demandée (DD), les phases 2 et suivantes de la primitive AUM (saut à l'étape 20) sont immédiatement appelées après que la dimension demandée (DD) a été chargée dans le registre GRSZ.
- 16) Si l'indicateur SW1 \neq 0, les informations sur la zone suivante sont extraites.
- 17) Si la zone suivante est un segment échangeable ESW, le pointeur PTRS est pointé sur ledit segment ESW (ce pointeur est utilisé au moment de déchargement pour décharger les segments ESW entre le pointeur et l'adresse absolue contenue dans le registre GRAA).
- 18) Si la zone extraite n'est pas un segment non éligible par l'échange, la nouvelle dimension du pseudo-trou est calculée en revenant à l'étape 8.
- 19) Si à l'étape 10, la valeur du pointeur PTRS est supérieure à la valeur du registre GRAA, il y a alors un ou plusieurs segments ESW à échanger. Ces segments sont ceux qui se trouvent entre l'adresse du registre GRAA et celle du pointeur PTRS. Ces segments sont déchargés et la base des données des segments est mise à jour, comme dans l'étape 3.
- 20) La primitive AUM est appelée (en sautant la séquence de mise en place des sémaphores).
- 21) Après attribution de la mémoire, le segment (Sx) est transféré à son nouvel emplacement.
- 22) Après mise à jour de tous les pointeurs du segment, la procédure RMM (libération de la mémoire principale) qui sera décrite ci-après est appelée pour libérer la zone précédemment attribuée au segment.
- 23) Le nouvel emplacement du segment est chargé dans le registre général GRAA et un code de retour RC5 (indiquant que l'emplacement du segment a été changé est transmis au demandeur).
- 6 - 2 - 3 - Attribution de la mémoire de commande (ACM).
- 6 - 2 - 3 - 1 - Description générale.
- Cette procédure comporte deux phases. Dans la phase 1, si la dimension du plus grand trou de la mémoire de commande (CMBHSZ) est

supérieure ou égale à la dimension demandée (DD), la liste des trous de la mémoire de commande CMHLLIST est explorée en commençant à partir de l'adresse la plus petite pour trouver un trou suffisamment grand pour satisfaire la demande d'espace de mémoire de commande. Lorsqu'un tel trou est
5 trouvé, la base des données des segments est mise à jour et un retour est transmis au demandeur avec un code de retour indiquant le succès de l'opération (RCO) et l'adresse absolue de la zone attribuée.

Dans la phase 2, si la dimension du plus grand trou est inférieure à la dimension demandée, un essai est effectué pour accroître la dimension de
10 la mémoire de commande en repoussant la limite en cours avec la mémoire des utilisateurs dans le sens des adresses croissantes dans l'espoir que la combinaison des trous de chaque côté de la limite produira un trou combiné résultant suffisamment grand pour satisfaire la demande.

Si cette tentative ne réussit pas, un retour est transmis au demandeur
15 avec un code de retour d'échec RC8.

Si l'essai réussit, la base des données des segments est mise à jour et un retour est transmis au demandeur avec le code de retour de réussite RC0 et l'adresse absolue de la zone attribuée.

6 - 2 - 3 - 2 - Détails du traitement

20 6 - 2 - 3 - 2 - 1 - Phase 1

1) Si la dimension du plus grand trou de la mémoire de commande est inférieure à la dimension demandée, la phase 2 est exécutée à partir de l'étape 11 ci-après.

2) Si la dimension du plus grand trou de la mémoire de commande
25 CMBHSZ est égale ou supérieure à la dimension demandée, il existe un trou suffisant dans la mémoire de commande pour satisfaire la demande. La liste des trous de la mémoire de commande CMHLLIST est explorée en partant des adresses les plus basses. L'adresse du dernier trou de la mémoire de commande (CMLHADR) est extraite de la table COM TAB.

3) Chaque dimension de trou est comparée à la dimension demandée. Si la dimension d'un trou est inférieure à la dimension demandée, l'adresse du trou suivant dans la liste CMHLLIST est extraite. Si la dimension d'un trou est égale ou supérieure à la dimension demandée (DD) mais inférieure à $DD + 32$, la totalité du trou est attribuée au demandeur et une nouvelle en-
35 tête de trou est créée. Le trou est alors enlevé de la liste CMHLLIST et les pointeurs de trous sont mis à jour dans les trous adjacents.

4) Si la dimension du trou est supérieure ou égale à la dimension demandée + 32 ($DD+32$), une partie de l'ancien trou est attribuée au demandeur et un nouveau trou est créé avec la partie restante. La dimension du nouveau
40 trou est calculée comme suit :

Dimension du nouveau trou = ancienne dimension - (DD + 32).

5) La liste des trous de la mémoire de commande CMHLLIST est à nouveau mise à jour: une nouvelle en-tête de trou est créée et les pointeurs de trous dans les en-têtes précédant et suivant le nouveau trou et dans celle du nouveau trou sont mis à jour.

6) Les pointeurs de zone sont mis à jour dans l'ancienne et la nouvelle en-tête de trou et dans l'en-tête de la zone suivante.

7) Le type de zone de la zone attribuée au demandeur est mis à jour (ARTYP = 1 dans l'en-tête du trou cf, 5, 4, 3 ci-dessus).

8) La dimension du plus grand trou de la mémoire de commande CMBHSZ est comparée à la dimension de l'ancien trou: si elles sont égales la valeur de CMBHSZ de la table COM TAB doit être mise à jour.

9) Les zones suivantes de la table COM TAB sont mises à jour si nécessaire :

(1) CMFHADR (adresse du premier trou de la mémoire de commande) et (2) CMLHADR (adresse du dernier trou de la mémoire de commande).

10) L'adresse absolue de la zone attribuée est chargée dans le registre général GRAA et un code de retour réussi RC1 est chargé dans le registre général GRRC et un retour est transmis au demandeur.

6 - 2 - 3 - 2 - 2 - Phase 2

11) L'adresse de la première zone dans la mémoire des utilisateurs est extraite au moyen du pointeur UMCURADR.

12) Le type de la zone précédente est vérifiée (cette zone est la dernière zone de la mémoire de commande). Selon le type de zone, une zone de travail DIM est initialisée avec la valeur DIM égale à -32 (pour un segment de commande) et à la dimension du trou pour un trou.

13) Le type de la première zone de mémoire des utilisateurs est vérifiée. Si c'est un trou la valeur DIM est mise à jour comme suit : nouvelle dimension = ancienne dimension + 32 + dimension du trou. Si la zone lue n'est pas un trou, un retour est transmis au demandeur avec un code de retour d'échec RC2.

14) La nouvelle valeur DIM est comparée à la dimension demandée si la valeur DIM est inférieure à la dimension demandée, un retour est transmis au demandeur avec un code de retour d'échec RC2. Cependant, si la valeur DIM est supérieure ou égale à la dimension demandée, il existe un espace suffisant pour satisfaire la demande.

15) Le trou en cause dans la mémoire des utilisateurs est retiré de la liste (UMHLLIST) et le pointeur du trou suivant dans l'en-tête du trou précédent est mis à jour.

16) Les informations ci-après sont mises à jour dans la table des

communications : UMAVLSZ, UMBHSZ (si nécessaire), UMBPHSZ (si nécessaire), UMBPHADR (si nécessaire) et UMCURADR.

17) Le type de la zone précédente est vérifié (cette zone est la dernière de la mémoire de commande). Si c'est un segment de commande, le nouveau trou est ajouté à la liste des trous de la mémoire de commande (CMUHLIST). Si c'est un trou, les deux dimensions des trous sont ajoutées et les pointeurs de zone sont mis à jour ainsi que les informations ci-après dans la table COM TAB: CMAVLSZ CMBHSZ CMFHADR (si nécessaire) et CMLHADR (si nécessaire).

18) Un retour est alors transmis au demandeur.

6 - 2 - 4 - Libération de la mémoire principale (RMM).

6 - 2 - 4 - 1 Description générale.

La zone précédemment occupée par le segment (appartenant soit à la mémoire de commande soit à la mémoire des utilisateurs) est insérée dans la liste des trous. Le segment est enlevé de la liste à laquelle il appartenait et les diverses zones de la table des communications (COM TAB) sont mises à jour.

6 - 2 - 4 - 1 - 1 - Détails du traitement.

1) L'adresse absolue du segment, transmise en tant que paramètre par l'intermédiaire du registre GRAA est comparée à l'adresse en cours de la mémoire des utilisateurs (UMCURADR) pour déterminer si le segment appartient à la mémoire de commande ou à la mémoire des utilisateurs.

2) Si le segment réside dans la mémoire des utilisateurs, il est retiré de la liste à laquelle il appartient (ESWLIST ou NESWLIST). Si le segment était le premier ou le dernier de ces listes, la table des communications est mise à jour (FSGADR, LSGADR, FESWADR).

3) L'adresse de la zone suivante est extraite de l'en-tête du segment.

4) La dimension réelle du trou est calculée.

5) Le registre GRAA désigne alors l'en-tête de l'ancien segment et le type de zone et les autres attributs contenus dans cette en-tête sont vidés. La dimension est mise en mémoire dans l'en-tête du nouveau trou.

6) La quantité de mémoire disponible est accrue de la dimension DIM + 32.

7) L'adresse de l'en-tête de l'ancien segment est comparée à l'emplacement du premier trou de la partie correspondante de la mémoire principale (le premier trou est celui de la liste qui a la plus grande adresse absolue).

8) La position de la zone précédente est comparée à la position du premier trou.

9) Les deux trous sont réunis et la dimension est mise à jour, le

type de zone de l'ancien segment est remis à 1 et les zones, liaisons, liaisons de trous ainsi que l'adresse UMLAADR sont mises à jour.

10) L'ancien segment devient le premier trou dans la liste et la dimension du plus grand trou est mise à jour dans la table des communications, si nécessaire.

11) Lorsque le segment appartenait à la mémoire des utilisateurs, la dimension et l'adresse du plus grand pseudo-trou sont mises à jour, si nécessaire.

12) Les informations sur le trou suivant sont extraites (dans l'ordre des adresses croissantes). Si le trou suivant est adjacent à l'ancien segment, les deux trous sont réunis et la dimension est mise à jour dans l'en-tête de l'ancien segment. Les liens des trous sont mis à jour dans les en-têtes des trous. L'adresse UMLAADR est mise à jour, si nécessaire.

13) Si le trou trouvé était le premier de la liste, l'adresse contenue dans la table des communications devient l'adresse contenue dans le registre GRAA.

14) Le nouveau trou est inséré dans la liste des trous. Si le trou suivant est le dernier, l'ancien segment devient le nouveau dernier trou et l'adresse du dernier trou est mise à jour dans la table des communications.

15) Le nouveau trou est alors réuni ou lié au trou suivant.

6 - 3 - Gestion des segments.

6 - 3 - 1 - Chargement des segments (SGLD)

6 - 3 - 1 - 1 - Description générale

Cette procédure ne comporte qu'un point d'entrée pour une demande synchrone d'un processus demandeur (anneau 0) du système. Pour les demandes asynchrones d'un processus d'utilisateur, la procédure de chargement de segment est appelée par l'intermédiaire d'un processus du système.

Après que le descripteur du segment direct (DSD) du segment qui doit être écrit dans la mémoire principale a été extrait, les zones P, I et U du descripteur DSD sont vérifiées. Selon leur valeur, l'état du segment est déduit (cf Fig. 13).

Si le segment est présent, un retour est fait au demandeur; si le segment est instable la demande est différée, et si le segment est éligible pour être échangé, il est transféré à la liste des segments non éligibles pour l'échange (NESWLIST) et enfin si le segment est manquant, il est déplacé de la mémoire auxiliaire (en utilisant la primitive Mouvement segment à mémoire principale (MVMAIN) de la composante de gestion de la mémoire auxiliaire) après que l'attribution de mémoire des utilisateurs a été effectuée en utilisant la primitive AUM de la composante de gestion de la mémoire principale).

40 6 - 3 - 1 - 2 - Détails du traitement.

1) Lorsque la procédure de chargement du segment est entrée, l'identification du segment à écrire dans la mémoire principale est chargée dans le registre GRJPS et le type de demande dans le registre GRTY.

2) Le descripteur du segment spécifié est extrait.

5 3) Si le descripteur est indirect le descripteur direct est extrait.

4) Les valeurs P, I, U du descripteur direct DSD sont vérifiées.

5) Si $P = 0$, $I = 1$, $U = 0$ le segment est à l'état instable et la demande est différée.

10 6) Si $P = 0$, $I = 1$, $U = 1$, le segment est dans la liste des segments éligibles pour être échangés (ESWLIST). Le segment est transféré à la liste NESWLIST, la valeur de l'ensemble de travail instantané du groupe de processus (IWS) est accrue et un retour est transmis au demandeur après que les attributs du segment aient été mis à jour.

15 7) Si $P = 1$, $I = 0$, le segment est présent et un retour est transmis au demandeur.

8) Si $P = 0$, $I = 0$, le segment est manquant.

9) La détection d'emballlement est effectuée et s'il s'est produit un emballlement, il est supprimé en utilisant une fonction détaillée qui sera décrite au paragraphe 6, 3, 12, 2 ci-après.

20 10) L'indicateur CONTOURNEMENT dans le registre GRTY est vérifié. Si le demandeur est un processus de gestion de mémoire virtuelle, la sous-phase de réservation des zones du programme de canal CPAREA (étapes 11 et 12 ci-après) et la sous-phase de verrouillage est sautée. Dans tous les cas, le segment est mis à l'état instable ($P = 0$, $I = 1$, $U = 0$).

25 11) Le compte du compteur E/S de la mémoire auxiliaire (BSIO CNT) est comparé à la valeur maximale E/S de la mémoire auxiliaire (BSIOMAX). Si ces deux valeurs sont égales, cela signifie que toutes les zones de programme de canal sont utilisées. Dans ce cas, le compte du compteur des processus en attente MMWPRCT est accru d'une unité et la demande est
30 différée (mise en attente sur le sémaphore SYSMMSAT).

12) S'il existe au moins une zone de programme de canal CPAREA libre, le compte du compteur E/S de la mémoire auxiliaire (BSIOCNT) est accru d'une unité, une entrée de programme de canal est choisie et l'adresse absolue de ce programme de canal (CPAREA) est placée dans le registre
35 GRCPAA.

13) La sous-phase de verrouillage est déclenchée (cf 6, 3, 12, 3 ci-après).

14) Si le segment est privé et réservé à un groupe de processus (c'est-à-dire que le bit indicateur PARTAGE dans le mot 5 de l'en-tête
40 du segment est différent de 0) l'ensemble de travail instantané IWS du

groupe de processus correspondant est mis à jour et la procédure d'attribution de mémoire des utilisateurs UAM est alors déclenchée.

5 15) Si le demandeur est extérieur au complexe de gestion de mémoire virtuelle (ce qui est déterminé en vérifiant le registre GRTY) la commande exclusive du sémaphore SYSALL est libérée par une opération V-op et la sous-phase de déverrouillage est déclenchée.

10 16) La dimension du segment est alors comparée à la dimension d'un bloc et si elle est plus grande, la procédure MVMAIN (mouvement à la mémoire principale) est appelée autant de fois qu'il est nécessaire. Dans ce cas, les premières procédures MVMAIN sont du type 2 (MVMAIN 2) et la dernière est du type 1 (MVMAIN 1) (pour plus de détails on se référera au paragraphe 6, 4, 3 ci-après).

15 17) La dimension maximale du segment (MSIZE dans le mot 7 de l'entête) est comparée à la dimension la plus grande possible MAXMAXSZ de la table COM TAB. Si la dimension MSIZE est supérieure à MAXMAXSZ, le processus qui a émis la demande est avorté.

18) Dans la sous-phase de sortie, la zone de programme de canal CPAREA utilisée pour le programme de canal est tout d'abord libérée.

20 19) Si la mémoire des utilisateurs n'est pas saturée à ce moment (IOSTOP = 0) et s'il y a un ou plusieurs processus en attente sur le sémaphore SYSMMSAT (MMWPRCT \neq 0), le compte du compteur MMWPRCT est diminué d'une unité et une opération V-op est effectuée sur le sémaphore SYSMMSAT pour indiquer qu'une zone de programme de canal CPAREA a été libérée.

25 20) Si la mémoire des utilisateurs est saturée (IOSTOP = 1) et si la nouvelle valeur du compteur d' E/S de la mémoire auxiliaire (BSIOCNT) est zéro, tous les segments de la mémoire des utilisateurs sont déverrouillés et la dernière phase de la primitive AUM (échange général et compression de la mémoire) peut être exécutée. Alors, tout processus qui avait présenté antérieurement une demande AUM non satisfaite du fait de la saturation de la
30 mémoire des utilisateurs est réactivé par une opération V-op sur le sémaphore SYSBSIO.

21) Les attributs du segment sont mis à jour et un retour est alors transmis au demandeur.

35 6 - 3 - 2 - Déchargement des segments (SGUNLD).

6 - 3 - 21 - Description générale

Cette fonction permet à l'utilisateur de spécifier les segments qu'il n'est pas nécessaire de conserver en mémoire principale. La fonction peut être appelée avec une option "indicatif" ou "obligatoire" (bit 23 du registre
40 GRTY).

a) lorsque la fonction est appelée à titre "indicatif" le segment reste dans la mémoire des utilisateurs mais il devient éligible pour être échangé (c'est-à-dire qu'il est transféré de la liste NESWLIST à la liste ESWLIST). En outre, le compteur d'occupation d'espace du propriétaire est diminué de cette dimension de segment. Un segment ESW quitte la liste ESW et redevient un segment non éligible NESW dès qu'il est référencé ou que la fonction de chargement de segment est appelée pour ledit segment.

b) lorsque la fonction est appelée à titre "obligatoire" le segment est effectivement transféré de la mémoire des utilisateurs à la mémoire auxiliaire. Pour pouvoir appeler la fonction obligatoire, le demandeur doit être une mise à jour dynamique ou une partie de la gestion de mémoire virtuelle (comme indiqué par le registre GRTY).

6 - 3 - 2 - 2 - Détails du traitement

Compte tenu de la complexité des procédures de déchargement des segments, un ensemble de 9 organigrammes correspondant à cette procédure a été représenté sur les Fig. 18A à 18I. Sur ces fig., les chiffres entre parenthèses correspondent aux numéros des étapes décrites ci-après :

1) cf fig. 18A. Les paramètres d'entrée sont chargés dans les registres ^{généraux} (GRTY pour le type de demande, GRJPS pour l'identification du segment) de sorte qu'aucune exception de segment manquant ne peut se produire ultérieurement.

2) le numéro d'anneau du demandeur (trouvé dans un registre de base) est vérifié.

3) Si le demandeur a effectué son appel sur l'anneau 0, le bit indicateur SPEC est mis à 0 dans le registre GRTY du fait que la procédure n'a pas été appelée par son point d'entrée spécial et si le type indique que le segment est identifié par son adresse absolue dans le descripteur de segment direct, (DSD) le contenu du registre GRJPS est chargé dans le registre GRAA.

4) Si le demandeur a effectué son appel sur les anneaux 2 ou 3, le demandeur peut uniquement spécifier si sa demande est obligatoire ou informative (les autres zones du registre GRTY doivent être des zéros). Le segment à décharger ne peut pas appartenir à une table de segments de type 0 si le demandeur a effectué son appel sur les anneaux 2 ou 3 et le segment doit appartenir à l'espace d'adresses du demandeur.

5) Le demandeur entre alors la section critique et la commande exclusive de la base des données des segments est prise par une opération P-op sur le sémaphore SYS1 afin de manipuler les en-têtes des segments, les descripteurs de segments et la table des communications.

6) La fonction de déchargement de segment peut alors être entrée à ce

point à partir d'un processus du système, l'indicateur SPEC du registre GRTY étant tout d'abord mis à l'état 1.

7) cf Fig. 18B. Les paramètres d'entrée sont mis en réserve dans la pile de l'anneau 0 (du fait qu'il peut être nécessaire à une étape ultérieure de rétablir le contexte dans le cas où le processus doit être repris dès le début après que la commande exclusive du sémaphore SYS1 a été libérée).

8) Le descripteur du segment identifié soit par les tables J, P le numéro de table le segment (STN), l'entrée de table des segments (STE) soit par son adresse absolue, est extrait.

9) Les bits P, I du descripteur de segment sont vérifiés et interprétés comme représenté sur la fig. 13.

10) Si le descripteur de segment est indirect, le descripteur direct DSD est accédé par l'intermédiaire du descripteur indirect IDS.

11) Si le segment est manquant ou est un segment ESW et que la demande est indicative, un code de retour réussi est émis et le retour est transmis (par D7 Fig. 18G).

12) Si le segment est à l'état instable, l'horloge est appelée et la requête est différée.

13) Si l'entrée de la table des segments spécifiée est une entrée réservée ou décrit un descripteur de segment fantôme, la demande est invalide et la demande est avortée (par E7 fig. 18G).

14) Si le segment est présent dans la mémoire principale (éligible ou non éligible pour être échangé (ESW ou NESW), le segment est vérifié pour s'assurer qu'il est situé dans la mémoire des utilisateurs échangeables, le registre GRAA est chargé pour désigner l'en-tête du segment, le registre GRJPS est chargé avec la totalité de l'identification du segment telle qu'elle est trouvée dans l'en-tête et il est déterminé si le segment est actuellement résidant auquel cas la demande est différée.

15) cf Fig. 18C. Le segment est rendu manquant en remettant le bit de présence P à zéro.

16) Si la demande est indicative, le segment est retiré de la liste NESWLIST (les compteurs et les liens sont mis à jour dans les en-têtes et dans la table COM TAB et le bit indicateur ESW est mis à l'état 1 dans l'en-tête).

17) L'ensemble de travail instantané est diminué de la dimension du segment si le segment n'est pas du type 0.

18) Le profil classique ESW est inscrit dans le descripteur du segment (bit U = bit I = 1).

19) La zone de dimension dans le descripteur de segment direct (DSD) est ajustée.

20) Si la demande est obligatoire, la dimension du segment (telle que trouvée dans le descripteur direct DSD) est chargée dans le registre GRSZ et ajustée.

5 21) Les zones de dimensions maximales et d'adresse relative sur le disque de l'en-tête sont chargées dans les registres GRMSZ et GRDA.

22) Il est déterminé si une copie du segment existe déjà dans la mémoire auxiliaire.

10 23) S'il a été écrit dans le segment (bit W dans le descripteur DSD = 1) ou si le segment a été modifié (EXP dans le descripteur DSD = 1), il doit être ^{re}transféré dans la mémoire auxiliaire; sinon l'espace qu'il occupait dans la mémoire des utilisateurs est simplement libéré étant donné que la copie dans la mémoire auxiliaire est toujours valide.

15 24) (cf Fig. 18D) l'indicateur du système IOSTOP (dans la zone d'alignage SW2 de la table COM TAB) est vérifié pour déterminer si une nouvelle opération d'entrée/sortie peut être entreprise avec la mémoire auxiliaire.

25) Le compteur des opérations E/S simultanément en cours (BSIO-CNT) est comparé à sa limite supérieure préétablie (BSIOMAX).

20 26) Si une nouvelle opération E/S peut être entreprise, une zone libre de programme ^{de}canal (CPAPEA) est attribuée pour le compte du demandeur et les structures correspondantes sont mises à jour.

27) Le segment est indiqué instable (bit U = 0, bit I = 1) tant que le transfert du segment à la mémoire auxiliaire n'est pas achevé.

25 28) Le registre GRAA pointe alors le premier multiplet de l'en-tête qui doit être écrit dans la mémoire auxiliaire et la procédure se poursuit à l'étape 30.

30 29) Si aucune nouvelle opération E/S ne peut être entreprise à ce moment (cf étapes 24 et 25 ci-dessus), l'état initial est restauré dans le descripteur de segment direct (c'est-à-dire que le segment est rendu à nouveau présent) le compte du compteur du système des processus en attente (MMWPRCT) qui sont empêchés d'entreprendre de nouvelles opérations E/S est accru, la commande exclusive de la base des données des segments (SYS1) est momentanément libérée tant que le demandeur attend sur le sémaphore (sans message) SYSMMAT et les paramètres sont restaurés dans les
35 registres généraux. La procédure est alors reprise à l'étape 8.

30) (cf Fig. 18E) A la suite de l'étape 28, une vérification du registre GRDA est à nouveau effectuée pour déterminer si un espace a déjà été attribué au segment dans la mémoire auxiliaire.

40 31) Pour le cas où la fonction ABSM (attribution de la mémoire auxiliaire) est appelée plusieurs fois successivement, l'indicateur SW1

est mis à l'état 1 pour identifier le premier appel.

32) Du fait que le contenu du registre GRSZ est modifié au retour de la fonction ABSM, la quantité qu'il contient est mise en mémoire au préalable.

5 33) Si la mémoire auxiliaire est saturée, l'état initial est restauré dans le descripteur de segment et une vérification est effectuée pour déterminer si un code de retour doit être transmis au demandeur (par exemple si le demandeur est le chargeur de la mémoire virtuelle) ou si la demande du propriétaire du segment doit être avortée.

10 34) Lorsque l'indicateur SW1 est à l'état 1, il correspond à la première attribution d'espace pour le compte du segment spécifié. Dans ce cas, le compte des blocs attribués (SGNALU) est mis à l'état initial 1 dans son en-tête, l'adresse du disque du commencement du segment est mise en mémoire dans l'en-tête et l'attribut d'expansion est remis à zéro.

15 35) Lorsqu'un espace supplémentaire est attribué au segment spécifié, le compte des blocs attribués est accru d'une unité en préparation de l'appel de la fonction (Mouvement de retour à mémoire auxiliaire) MVBAC (comme représenté sur la Fig. 18H) afin de mettre en forme un bloc et d'écrire dans l'enregistrement R0 de la première piste, l'adresse sur le disque du bloc
20 suivant.

36) S'il est besoin de plus d'espace et que la fonction ABSM doit être rappelée, l'adresse en cours du disque est sauvegardée dans le registre GRRO avant que la fonction MVBAC soit appelée afin de mettre en forme la dernière partie du segment dans la mémoire auxiliaire. On notera que, du fait
25 que la dimension du segment a été modifiée depuis la dernière fois qu'il a été copié dans la mémoire auxiliaire. Le segment doit être réécrit bloc par bloc jusqu'à ce que le nombre de blocs attribué au segment soit trop petit. Ensuite, de nouveaux blocs doivent lui être attribués.

30 37) Si le segment ne peut pas être déchargé par suite de la saturation de la mémoire auxiliaire, le segment est effacé de la mémoire des utilisateurs et le groupe de processus propriétaire du segment est avorté par le processus de mise en séquence.

38) (cf Fig. 18F) lorsqu'un espace suffisant est attribué dans la mémoire auxiliaire, le segment est réécrit bloc par bloc. A chaque opération de
35 Mouvement à la mémoire auxiliaire MVBAC, l'adresse absolue du bloc suivant est obtenue en lisant le contenu de l'enregistrement R0. Enfin, la fin du segment est écrite dans le dernier bloc.

39) La combinaison "stable" et "manquant" est alors inscrite dans le descripteur de segment direct.

40 40) L'adresse du disque est sauvegardée du fait que, lorsque le

segment devient un trou, la zone correspondante de l'en-tête est remplacée par les liens de trous.

41) L'ensemble de travail instantané est diminué si le segment qui quitte la mémoire des utilisateurs n'est pas partagé et n'est pas éligible pour être échangé.

42) L'espace est libéré dans la mémoire principale et le descripteur de segment direct DSD est réécrit de telle sorte que l'adresse relative du disque (CHR) est mise en mémoire dans la zone de base et la zone de dimension est augmentée d'une unité si le segment échangé n'était pas éligible pour être échangé.

43) (cf Fig. 18G) le compte des opérations E/S en cours avec la mémoire auxiliaire est diminué d'une unité et la zone du programme^{de} canal est rendue du pool commun et son entrée correspondant dans le plan binaire est mise à jour dans la table des communications.

44) Si l'indicateur du système IOSTOP est à l'état 1, cela signifie que la mémoire des utilisateurs est entrée dans l'état de saturation. Dans ce cas, le processus qui détecte cet état de saturation attend jusqu'à ce que toutes les opérations E/S en cours avec la mémoire auxiliaire soient achevées. Ainsi, un test est alors effectué pour déterminer si l'opération E/S qui vient de s'achever est la dernière. Dans ce dernier cas, le processus qui a détecté la saturation est réactivé.

45) La commande exclusive de la base des données des segments est libérée et le processus sort de la section critique.

6 - 3 - 3 Changement de la dimension des segments (SGSIZE).

25 6 - 3 - 3 - 1 - Description générale

Cette procédure peut être entrée soit à partir d'un processus de la gestion de mémoire soit à partir d'un processus d'utilisateur.

Le descripteur de segment direct est extrait et selon la valeur des zones P, I, U et W, l'état du segment est déduit (cf fig. 13). Si le format du descripteur de segment direct est invalide, le processus est tué. Si le segment est instable, la demande est différée, s'il est manquant, le segment est chargé et s'il est dans la liste ESW échangeable, il est transféré dans la liste NESW (non échangeable). Une fois que le segment est présent dans la liste NESWLIST, la procédure du changement d'attribution de la mémoire des utilisateurs (CUM) est appelée.

Cependant, si la dimension demandée est zéro, le segment est transformé en segment fantôme (sous réserve qu'il n'ait pas de copie dans la mémoire auxiliaire) et l'espace attribué au segment est libéré par la procédure RMM (libération de la mémoire principale).

40 Si le segment est un segment fantôme, il est transformé

en segment réel soit dans la mémoire de commande soit dans la mémoire des utilisateurs selon l'attribut CONT (cf Fig. 9).

Cependant, une fois que l'ensemble de travail déclaré dans la mémoire est dépassé, le processus est avorté (si le demandeur était un processus d'utilisation) ou un code de retour "ensemble de travail dépassé" est transmis au demandeur (s'il s'agit d'un processus du système).

6 - 3 - 3 - 3 - Détails de traitement

1) Lors de l'entrée de la procédure par un processus d'utilisateur, l'identification du segment et la nouvelle dimension demandée sont chargés dans les registres généraux GRJPS et GRSZ.

2) La validation des paramètres est effectuée selon le numéro d'anneau du demandeur. Si le numéro d'anneau du demandeur (CRN) est égal à zéro, il n'y a pas de validation mais un traitement complémentaire des paramètres d'entrée est effectué (du fait de la souplesse la plus grande accordée aux utilisateurs de l'anneau 0) de la façon suivante; le paramètre d'entrée TYPE est mise en mémoire dans le registre RGTY, l'indicateur SPECIAL est mis à zéro et, si l'identification dans le registre GRJPS est l'adresse absolue du descripteur du segment direct DSD, elle est transférée au registre général GRAA.

Si le numéro d'anneau du demandeur CRN est égal à 1, le paramètre TYPE est validé, les zones STN, STE du registre GRJPS sont remplis de 1 et le descripteur du segment est validé.

Si le numéro d'anneau du demandeur est égal à 2 ou 3, le paramètre d'entrée TYPE (qui n'est pas visible pour l'utilisateur) est rempli de 0 et le type du segment est contrôlé. Si le segment appartient à une table de segments de type 0, le processus est avorté.

3) La commande exclusive des sémaphores du système SYSALL et SYS1 est prise. SYSALL est le sémaphore qui attribue la mémoire des utilisateurs à un utilisateur à la fois afin d'éviter les interactions entre plusieurs demandeurs.

4) Le processus du système entre la procédure SSIZE à ce point auquel cas la commande exclusive des sémaphores SYSALL et SYS1 a déjà été prise.

5) L'indicateur SPECIAL est mis à 1 pour les processus du système dès leur entrée dans la procédure SSIZE.

6) Le descripteur du segment spécifié est extrait.

7) Si le numéro d'anneau du demandeur est supérieur au numéro d'anneau d'écriture du descripteur du segment, le processus est avorté (il doit être possible d'écrire dans le segment).

8) Si le descripteur du segment est indirect, le descripteur de

segment ^{direct} est extrait.

9) Les zones P, I, U, W du descripteur de segment direct sont contrôlées. Si P = 0; I = 1, U = 0, le segment est à l'état instable et la demande est différée. Si P = 0, I = 0, U = 1 un format invalide est indiqué et le processus est avorté.

10) Pour les autres combinaisons (Présent, ESW, Manquant, Fantôme) la sous-phase de réservation CPAREA et le même traitement (tel que décrit au paragraphe 6, 3, 1, 2 pour la procédure de chargement de segment SGLD) est effectué.

11) Si la demande concerne un segment fantôme, la dimension demandée est comparée à la dimension maximale déclarée et la dimension maximale est comparée à la dimension la plus grande possible.

12) Si la demande concerne la mémoire de commande (Attribut CONT = 1) selon le type de segment, l'ensemble de travail correspondant est accru de la dimension demandée et comparé à l'ensemble de travail déclaré de la façon suivante :

Si l'ensemble de travail déclaré est dépassé, il est mis fin au groupe de processus demandeur (s'il s'agit d'un processus d'utilisateur) et un code de retour RC3 est transmis au demandeur (s'il s'agit d'un processus du système).

Si l'ensemble de travail déclaré n'est pas dépassé, la procédure d'attribution de mémoire de commande est appelée. Si la procédure d'attribution ne réussit pas, le programmeur est informé et le processus est suspendu (s'il s'agit d'un processus d'utilisation) après retrait d'attribution de la zone de programme de canal (CPAREA). Si l'attribution réussit, l'ensemble de travail correspondant est mis à jour, l'en-tête de segment est remplie avec les attributs du segment et le descripteur du segment direct est mis à jour. Le retour est alors fait au demandeur par l'intermédiaire de la sous-phase de sortie.

13) Si la demande concerne la mémoire des utilisateurs (attribut CONT = 0), selon le type de segment, l'ensemble de travail instantané (IWS) est mis à jour, le segment est mis à l'état instable et la procédure d'attribution de la mémoire des utilisateurs est appelée. Après achèvement de l'attribution de la mémoire (AUM), la commande exclusive du sémaphore SYSALL est libérée et le descripteur direct DSD est mis à jour. Ensuite, la sous-phase de sortie est entrée.

14) Si le segment est manquant, la procédure de chargement de segment (SGLD) est appelé avec l'option CONTOURNEMENT.

15) Si le segment est dans la liste NESW (non échangeable), il passe à l'état instable.

16) Si le segment est dans la liste ESW (échangeable), il est transféré à la liste NESW et l'accroissement de dimension est calculé.

17) Si la dimension demandée est zéro et si le segment n'a pas de copie dans la mémoire auxiliaire, le segment est transformé de segment réel en segment fantôme de la façon suivante: la commande exclusive du sémaphore SYSALL est libérée, le descripteur du segment fantôme est construit en partant des informations contenues dans l'en-tête du segment, la procédure de libération de la mémoire principale (RMM) est appelée, l'ensemble de travail instantané (IWS) correspondant est mis à jour, l'entrée/sortie physique est notifiée (si le segment est une mémoire tampon) et un retour est transmis au demandeur par l'intermédiaire de la sous-phase de sortie.

18) Si la demande concerne un segment réel dans la mémoire des utilisateurs, l'ensemble de travail instantané (IWS) est mis à jour (selon le type de segment).

19) La procédure de changement de la mémoire des utilisateurs (CUM) est appelée.

20) Si le segment a été translaté, la zone de dimension dans le descripteur direct DSD est mise à jour.

21) Si la dimension demandée était zéro, le segment est transféré de la liste NESWLIST à la liste ESWLIST et le descripteur DSD est mis à jour (dans ce cas, seule l'en-tête du segment continue d'exister).

22) Si la nouvelle dimension du segment n'est pas nulle, le descripteur DSD est mis à jour.

23) Un retour est transmis au demandeur par l'intermédiaire de la sous-phase de sortie.

6 - 3 - 4 - Modification d'attributs de segments (SGATTR)

6 - 3 - 4 - 1 - Description générale

L'objet de cette primitive est soit de lire soit de modifier les attributs d'un segment qui se trouvent soit dans un descripteur de segment direct soit dans une en-tête de segment.

Si la fonction est appelée avec l'option "LECTURE" (indiquée par le mot 5, bit 21 du registre GRTY), l'ensemble du contenu du descripteur de segment direct est mis en mémoire au début de la zone de réception du demandeur. Si alors le segment spécifié est un segment, le contenu de l'en-tête est mis en mémoire dans la zone de réception du demandeur.

Si la fonction est appelée avec l'option MISE EN PLACE D'ATTRIBUT, les zones sont modifiées sélectivement selon un masque d'attributs construit par le demandeur. Les règles standard sont appliquées pour contrôler le droit du demandeur à modifier les zones. Si le demandeur est un utilisateur général, le droit de modifier les droits d'accès dans le descrip-

teur DSD lui est refusé. Cependant, si le demandeur est un utilisateur du système (anneau 0 ou 1) il peut modifier le coefficient de résidence qui est contenue dans l'en-tête. Chaque fois la modification demandée n'est pas en accord avec les règles établies, le processus du demandeur est avorté.

5

6 - 3 - 4 - 2 Détails du traitement

1) Les paramètres d'entrée sont chargés dans les registres pour empêcher toute nouvelle modification.

2) Si le demandeur exécute la procédure dans l'anneau 0, il peut spécifier l'adresse absolue du descripteur de segment pour identifier le segment en cause.

10

3) Le format du type de demande est validé.

4) L'identification donnée du segment est validée par le déclenchement du développement d'adresse.

5) Si l'option LECTURE est demandée, la longueur de la zone réceptrice est validée.

15

6) Si le demandeur demande la modification du coefficient de résidence, un contrôle est effectué pour s'assurer que le demandeur est un processus du système (le coefficient de résidence 0 ne peut être spécifié car il est réservé pour une utilisation spécifique du système).

20

7) La commande des deux sémaphores SYSALL et SYS1 est prise sur une base exclusive. Le sémaphore SYSALL permet l'accès à la primitive d'allocation de mémoire des utilisateurs (AUM), s'il devient nécessaire de charger le segment référencé et le sémaphore SYS1 permet la manipulation (AUM) de la base des données des segments.

25

8) Le descripteur du segment référencé est extrait.

9) Si le descripteur est un descripteur de segment indirect, le descripteur de segment direct est extrait.

10) Les diverses combinaisons classiques des zones du descripteur de segment sont vérifiées. L'indicateur SW est mis à l'état 1 pour distinguer le descripteur de segment fantôme des autres cas.

30

11) Si l'entrée de la table de segment référencée n'est pas un segment réel (c'est-à-dire qu'elle est un segment invalide ou un segment fantôme), la commande exclusive du sémaphore SYSALL est libérée par une opération V-op car aucun chargement n'est nécessaire.

35

12) Si le segment référencé est manquant, il est alors chargé dans la mémoire principale au moyen d'une procédure SGLD (chargement de segment).

13) Si le segment référencé est à l'état instable, la demande est suspendue jusqu'à ce que le segment soit revenu à l'état normal.

14) La commande exclusive du sémaphore SYSALL est libérée par une opération V-op dès qu'il est assuré que le segment est présent.

40

15) Si l'indicateur SW = 0, l'entrée spécifiée est un segment réel (cf étape 10 ci-dessus).

16) Si l'option LECTURE a été spécifiée, le contenu du descripteur DSD et, le cas échéant, le contenu de l'en-tête sont mis en mémoire dans
5 la zone de réception.

17) Si l'option MISE a été spécifiée, cette fonction n'est permise que pour un segment réel ou pour un descripteur de segment fantôme. Si les paramètres spécifiés sont invalides, le processus du demandeur est avorté. (Si nécessaire, la ressource du système protégée par le sémaphore SYS1
10 est préalablement libérée).

18) Le masque d'attributs est tout d'abord vérifié pour déterminer si le bit d'autorisation d'écriture, le bit d'autorisation d'exécution ou l'anneau d'écriture est spécifié. Dans ce cas, le numéro d'anneau du demandeur trouvé dans le registre de base d'anneaux BRO est comparé au minimum à la fois
15 de l'anneau de lecture (RDO) et de l'anneau d'écriture (WRO) qui existe dans le descripteur DSD à modifier.

19) Si l'anneau de lecture ou l'anneau maximal est spécifié, le numéro d'anneau du demandeur est comparé à la zone correspondante existant dans le descripteur DSD. Le demandeur doit avoir un anneau plus privilégié pour
20 modifier cette zone.

La nouvelle zone remplace l'ancienne, la mise à jour n'étant effectuée que si le nouvel anneau à mettre en place est supérieur ou égal au numéro/^{d'anneau} du demandeur. Si le coefficient de résidence est spécifié, il est mis en mémoire dans l'en-tête et la sortie est effectuée après la libération de la res-
25 source du système protégée par le sémaphore SYS1.

6 - 3 - 5 Mise du bit de disponibilité (SDAVL) et

6 - 3 - 6 Restauration du bit de disponibilité (SDUNAVL)

1) Le paramètre d'entrée identifiant le descripteur du segment est mis en mémoire dans le registre général GRJPS.

30 2) Si le numéro d'anneau du demandeur est égal à 1, les paramètres d'entrée sont validés.

3) La commande exclusive de la base des données des segments est prise par une opération P-op sur le sémaphore SYS1.

4) Le numéro d'anneau du demandeur est comparé au numéro d'anneau
35 d'écriture du descripteur du segment. Si le numéro d'anneau du demandeur est supérieur au numéro d'anneau d'écriture, le processus est avorté.

5) Selon la demande, le bit de disponibilité est mis à 1 ou remis à zéro, sauf dans le cas où l'état initial du bit de disponibilité correspond à l'état demandé; dans ce cas, un code de retour RC2 est transmis au deman-
40 deur. Si l'opération est achevée avec succès, le sémaphore SYS1 est

libéré et le retour est transmis au demandeur avec un code de retour de réussite RCO.

6 - 3 - 7 - Libération de la mémoire virtuelle par un groupe de processus (RELVM)

5 6 - 3 - 7 - 1 - Description générale

La mémoire principale est explorée en partant des adresses supérieures jusqu'à la limite noyau/mémoire de commande et toutes les zones attribuées à des segments appartenant au groupe de processus spécifié (numéro J) sont libérées. Ensuite, la procédure de libération de la mémoire

10 auxiliaire (RBSM) est appelée pour libérer l'ensemble de l'espace attribué au groupe de processus spécifié dans la mémoire auxiliaire.

6 - 3 - 7 - 2 - Détails du traitement

1) La commande exclusive des sémaphores du système SYSALL et SYS1 est prise.

15 2) L'adresse de la dernière zone de la mémoire des utilisateurs (UMLAADR) est extraite.

3) Si la zone dont l'adresse est extraite est un trou ou un segment n'appartenant pas au groupe de processus spécifié, l'adresse de la zone précédente est extraite.

20 4) Si le segment appartient au groupe de processus spécifié, son descripteur de segment direct DSD est extrait et l'état du segment est vérifié. Si le segment est instable la demande est différée. Si le segment est présent, la procédure de libération de la mémoire principale est appelée (RMM) et l'adresse de la zone précédente est extraite pour poursuivre l'opération de

25 libération.

5) Lorsque la limite noyau/mémoire de commande est atteinte, la procédure de libération de la mémoire auxiliaire (RBSM) est appelée et ensuite en retour est transmis au demandeur.

6 - 3 - 8 - Création de descripteur de segment indirect (ISDCRT)

30 1) Les paramètres d'entrée sont chargés dans les registres généraux. (Cette primitive ne peut être appelée que par les demandeurs (anneaux 0, 1) du système).

2) Le format de l'entrée de la table des segments est contrôlé. (Si l'anneau du demandeur est 1). Si l'entrée spécifiée n'a pas le format "vacant"

35 le processus du demandeur est avorté.

3) L'adresse absolue du descripteur du segment direct est extraite et mise en mémoire dans la zone d'emplacement du descripteur indirect ISD.

4) La configuration de bits caractérisant un descripteur indirect est mise en mémoire dans les quatre premiers bits du descripteur indirect ISD.

40 5) Les droits d'accès sont copiés dans le descripteur direct DSD.

6) Si le demandeur est la gestion des données, le bit de gestion des données du descripteur indirect ISD est mis à l'état 1.

7) Si le demandeur est la gestion des messages, le bit de gestion des messages du descripteur indirect ISD est mis à l'état 1.

5 8) Si le demandeur n'est ni la gestion des données ni la gestion des messages, les deux bits correspondants sont mis à zéro.

6 - 3 - 9 - Suppression du descripteur de segment indirect (ISDDEL)

1) Le paramètre d'entrée est chargé dans un registre général (cette primitive ne peut être appelée que par les demandeurs des anneaux 0, 1).

10 2) Si le numéro d'anneau du demandeur est égal à 1, un contrôle est effectué pour s'assurer que l'entrée spécifiée a le format d'un descripteur de segment indirect ISD et que le bit de gestion des données ou de gestion des messages est à l'état 1. Sinon la demande est avortée.

15 3) Si le format est valide, l'entrée spécifiée est transformée en entrée vacante.

6 - 3 - 10 - Mouvement de segment (SGMOVE)

6 - 3 - 10 - 1 - Description générale

Cette primitive permet à un segment (ou à une partie d'un segment) qui peut ne pas appartenir à l'espace d'adresses du demandeur d'être copié dans
20 un segment destinataire qui appartient à l'espace d'adresses du demandeur.

Après validation des paramètres d'entrée (pour un numéro d'anneau du demandeur égal à 1), les deux segments contenant la zone d'émission et la zone de réception sont chargés dans la mémoire des utilisateurs (si l'un ou l'autre est manquant). Puis, le transfert des données est effectué entre les
25 deux zones spécifiées de la mémoire principale.

6 - 3 - 10 - 2 - Détails du traitement

1) Les paramètres d'entrée sont mis en mémoire dans des registres généraux.

2) Si le numéro d'anneau du demandeur est égal à 1, les paramètres
30 d'entrée sont validés.

3) La commande exclusive des sémaphores du système SYSALL et SYS1 est prise.

4) Les segments de la zone d'émission et de la zone de réception sont chargés s'ils sont manquants. Ensuite, la commande exclusive du séma-
35 phore SYSALL (qui était nécessaire pour appeler la procédure de chargement de segment (SGLD) est libérée.

5) Des validations complémentaires, qui ne pouvaient être exécutées lorsque le segment correspondant était manquant, sont alors effectuées.

6) La dimension de la zone émettrice est comparée à la longueur du
40 transfert demandé. Si la longueur de transfert spécifiée dépasse la

dimension de la zone d'émission, seule l'information comprise entre le déplacement de la zone d'émission et la limite supérieure du segment d'émission est transférée (dans ce cas le paramètre de sortie SENDISP (déplacement du segment d'émission) est rempli de zéro). Dans l'autre cas, ce

5 paramètre est chargé avec le déplacement du premier multiplet dans le segment qui suit la zone transférée.

7) Le retour est transmis au demandeur.

6 - 3 - 11 - Déchargement de segment spécial (LDUNLD).

6 - 3 - 11 - 1 - Description générale

10 L'objet de cette fonction est de transférer de façon synchrone un segment emboîté qui est contenu à l'intérieur d'un autre (un segment "emboîteur") à la mémoire auxiliaire.

Le "segment" qui est effectivement déchargé est presque un segment effectif mais il n'apparaît cependant sur aucune liste de segment. Il s'agit

15 d'un segment transitoire qui réside provisoirement dans la mémoire des utilisateurs avant d'être écrit dans la mémoire auxiliaire et, en conséquence, une interface spéciale est nécessaire du fait que le groupe de processus d'utilisateur auquel le "segment" appartient n'est pas encore actif.

6 - 3 - 11 - 2 - Détails du traitement

20 1) L'identification de segment du "segment emboîteur" est mise en mémoire dans le registre général GRJPS.

2) La commande exclusive des deux ressources du système est prise dans un ordre défini par l'intermédiaire des sémaphores SYSALL et SYS1.

25 3) Une zone de programme de canal qui est nécessaire pour les opérations ultérieures E/S est réservée.

4) Si le segment "emboîteur" est manquant, il est tout d'abord transféré à la mémoire principale au moyen de la fonction SGLD (chargement segment).

30 5) Le segment "emboîteur" est rendu "instable" de sorte qu'il ne peut être échangé tant que le segment "emboîté" n'est pas effectivement déchargé.

6) Le segment emboîté est sorti de la mémoire principale et chargé dans la mémoire auxiliaire. La fonction de déchargement de segment est appelée (avec une option "CONSERVATION" de sorte que l'espace de mémoire des utilisateurs occupé n'est pas libéré après le déchargement).

35

7) L'état instable du "segment emboîteur" est supprimé et son état antérieur est rétabli.

8) La commande exclusive de la base des données des segments est libérée.

40 9) Un code de retour émis par la primitive de déchargement de

segment et la commande est renvoyée au demandeur.

6 - 3 - 12 - Autres fonctions

6 - 3 - 12 - 1 - Détection et suppression de l'emballement

Des moyens sont offerts par la procédure de chargement de segments (SGLD) pour détecter l'emballement et le supprimer. L'emballement apparaît lorsque le "taux de rotation" des échanges entre la mémoire Principale et la mémoire auxiliaire est supérieure à un seuil donné. Avant d'examiner le procédé utilisé pour le contrôle de l'emballement, certains termes doivent être définis.

10 Le compte d'emballement (THRCNT) est le compte d'un compteur résidant dans le noyau qui donne à tout moment le nombre de procédures de chargement de segment (SGLD) émises depuis un temps initial spécifié. Ce compteur est périodiquement remis à zéro.

La période de comptabilisation d'emballement "T" est une durée donnée utilisée pour déterminer le taux de rotation en cours (TR) qui est le rapport entre le compte du compteur d'emballement (THRCNT) et l'intervalle de temps correspondant ΔT . Ce rapport n'est calculé que lorsque l'intervalle de temps dépasse la période de comptabilisation d'emballement (T).

Trois modes de taux de rotation sont définis. Le "mode normal" lorsque le taux de rotation est inférieur à une valeur donnée appelée "seuil minimal". Le "mode d'emballement" lorsque le taux de rotation est supérieur à une valeur donnée ou "seuil maximal" et le "mode intermédiaire" lorsque la valeur du taux de rotation est comprise entre les deux seuils.

Le procédé spécifique de détection et de suppression d'emballement est exécuté de la manière représentée sur l'organigramme de la fig. 19 et décrite ci-après.

6 - 3 - 12 - 2 - Détails du traitement

1) L'heure du jour (TOD) disponible dans la base du système est chargée dans une zone de travail (H2) (heure actuelle).

2) Le compte d'emballement est accru d'une unité.

3) L'intervalle de temps ΔT entre le temps actuel et le temps initial de comptabilisation (précédemment mis en mémoire dans une zone de travail H1) est calculé.

4) Si cet intervalle de temps (ΔT) est inférieur à la période de comptabilisation d'emballement (T), la séquence ^{de détection} d'emballement est terminée.

5) Si cet intervalle de temps ΔT est supérieur à la période T, le taux de rotation est calculé et le temps de la période de comptabilisation est ré-initialisé.

6) Si le taux de rotation est supérieur au seuil maximal, le nom du groupe de processus qui a le rapport le plus élevé d'ensemble de travail

instantané (ensemble de travail déclaré (IWS/DWS) est extrait.

7) L'indicateur d'emballlement TSW1 est mis à l'état 1.

8) La commande exclusive de la base des données des segments est libérée, la sortie est faite de la section critique et le programmeur est informé du mode d'emballlement et du nom du groupe de processus en cause qui est mis en attente sur le sémaphore SYSTHR (où il sera remis en activité par le programmeur).

9) La section critique est entrée et la commande exclusive de la base des données est reprise.

10) Si au dernier calcul du taux de rotation TR, le mode d'emballlement existait, le programmeur n'est pas notifié du retour à la normale tant que le taux de rotation n'est pas redevenu inférieur au seuil minimal, ces deux niveaux sont utilisés pour éviter les oscillations autour d'un seuil unique.

11) Si le taux de rotation est inférieur au seuil minimal, une notification est adressée au programmeur.

6 - 3 - 12 - 3 - Sous-phase de verrouillage

Pour empêcher "le pompage des segments" les segments du demandeur qui sont nécessaires au temps de SORTIE (c'est-à-dire ceux contenant le point de retour et le segment de pile en cours) sont rendus résidants en mémoire principale, si les procédures AUM ou CUM doivent être appelées pour satisfaire la demande. Ainsi, il est assuré qu'ils ne seront ni échangés ni sortis de la mémoire principale par l'algorithme d'attribution. Après achèvement des procédures AUM ou CUM, ces segments sont déverrouillés.

6 - 3 - 12 - 3 - 1 - Détails du traitement

Si les segments RP (segment qui contient le point de retour) et ST (segment de pile en cours du demandeur) sont tous deux présents, ils sont rendus résidants (RES = 1). Si l'un est présent et l'autre est manquant le segment présent est rendu résidant et l'autre segment est chargé et rendu résidant. Si ces segments sont tous deux manquants, ils sont tous deux chargés et rendus résidants.

6 - 4 - Gestion de la mémoire auxiliaire

6 - 4 - 1 Attribution de la mémoire auxiliaire (ABSM)

6 - 4 - 1 - Description générale

L'objet de cette primitive est d'attribuer de l'espace dans le fichier de la mémoire auxiliaire, sur demande.

L'espace dans la mémoire auxiliaire est demandé pour un segment à la fois et est attribué séquentiellement dans le bloc en cours de remplissage. Lorsqu'il n'y a pas d'espace libre dans ce bloc, un nouveau bloc est attribué.

La fonction d'attribution peut s'appliquer à deux domaines (parties)

distincts de la mémoire auxiliaire selon une option (indicateur DL) spécifiée par demandeur dans le type de la demande. Ces deux parties sont les suivantes.

5 La partie 2 est réservée aux fins de la mise à jour dynamique et constitue une extension de la partie permanente (partie 1).

La partie 2 est constituée par un nombre entier des blocs consécutifs la zone BS2NAVLU (nombre de blocs disponibles) dans la partie relative à la mémoire auxiliaire de la table COM TAB est initialisée à la valeur précédente et diminuée d'une unité chaque fois qu'un nouveau bloc est nécessaire. Lors-
10 que le bloc en cours de remplissage est plein, le bloc immédiatement suivant est attribué. Lorsqu'il n'y a plus d'espace disponible, un code de retour d'échec est transmis au demandeur.

La partie 3 est réservée à la fonction d'échange et est constituée par un nombre entier de cylindres consécutifs. Chaque cylindre est divisé en un
15 nombre entier de blocs (par exemple 1, 2 ou 4). La dimension des blocs est la même dans la partie 2 et dans la partie 3 de la mémoire auxiliaire.

Le nombre total des blocs contenus dans la partie 3 est spécifié au moment de la création du système et est mis en mémoire dans la zone BS3-NU de la table COM TAB.

20 Le nombre de bloc actuellement disponible est conservé dans la zone BS3NAVLU de la table COM TAB. Les blocs sont attribués bloc par bloc.

Une table binaire appelée plan des blocs de la partie 3 de la mémoire auxiliaire (BS3MAPU) décrit l'état d'attribution de tous les blocs appartenant à la partie 3. Par convention "0" signifie occupé ou non disponible et
25 "1" signifie libre.

Deux zones sont associées à ce plan binaire. La zone NBW contenant la longueur du plan en blocs et la zone INDEX (un pointeur à l'intérieur du plan).

Lorsqu'un bloc doit être attribué, le compteur des blocs disponibles
30 est tout d'abord vérifié pour s'assurer qu'il reste au moins un bloc libre. Puis le plan est exploré mot par mot. La position binaire d'extrême gauche qui contient un bit 1 (c'est-à-dire bloc libre) est déterminée et identifie le bloc qui sera attribué. La position binaire, en partant du début du plan, identifie l'adresse relative sur les disques du bloc correspondant par rap-
35 port au début de la partie 3. Enfin, le bit est mis à zéro dans le plan afin d'indiquer que le bloc correspondant est occupé.

Afin d'accélérer la recherche du bloc à attribuer, la zone Index est chargée avec le numéro du mot, à l'intérieur du plan, qui contient la position binaire du dernier bloc attribué. La recherche est effectuée à par-
40 tir du mot désigné par le pointeur INDEX. Lorsque le pointeur atteint la

fin du plan, il est remis en position de départ avec le numéro du premier mot (organisation circulaire).

5 A la différence de ce qui est fait dans le cas de l'attribution de la partie 2 (où un code de retour d'échec est transmis au demandeur en cas de saturation), aucun retour n'est transmis au demandeur tant que la demande concernant la partie 3 n'est pas satisfaite. La différence réside en ce que, dans la partie 3, le volume d'espace disponible peut s'accroître lorsqu'un groupe de processus prend fin et que la fonction de libération de la mémoire auxiliaire est appelée.

10 Il en résulte que, lorsqu'une demande ne peut pas être immédiatement satisfaite, elle est placée dans une file d'attente sur un sémaphore du système (SYSBKSAT) jusqu'à ce qu'un bloc devienne disponible. Cependant, la tendance à la saturation est détectée et des actions sont entreprises avant que le fichier de la mémoire auxiliaire soit complètement surchargé.

15 Pour la partie 3, de la mémoire auxiliaire, la stratégie d'attribution peut être divisée en deux phases.

Dans la première phase, l'espace est attribué pour le compte d'un groupe de processus pour lequel la demande est présentée. Ceci se produit soit lorsqu'il est besoin d'espace pour la première fois (pour le groupe de
20 processus) soit chaque fois que l'espace déjà attribué au groupe de processus est insuffisant. Dans ce dernier cas, le segment en cause peut ne pas avoir encore une copie dans la mémoire auxiliaire ou il peut posséder une zone réservée qui est devenue trop petite.

L'espace est attribué bloc par bloc de mémoire auxiliaire, dans la
25 première phase. Des zones contenues dans le segment de commande du groupe de processus (PGCS) (sous-zone de la mémoire auxiliaire) contiennent les paramètres qui caractérisent le bloc qui est en cours de remplissage. Ces paramètres sont l'adresse dans la mémoire auxiliaire du groupe de processus (PGCSADR) donnant l'adresse relative sur le disque (C, H, R) de l'enregistrement suivant à écrire dans le bloc et la quantité de mémoire encore disponible dans le bloc. Cette dernière valeur est donnée en nombre de pistes disponibles (PGTAV) et en nombre de multiplats disponibles dans la première
30 de ces pistes (PGBAV).

D'autres zones qui se trouvent dans le segment de commande du groupe
35 de processus contiennent les paramètres qui caractérisent les blocs précédemment attribués à ce groupe de processus. Ces paramètres sont (a) le nombre des blocs déjà attribués (PGBSNALU) et (b) le plan de ces blocs (PGMAPU) qui est constitué par une table binaire similaire à celle de la table des communications déjà décrite. La signification de chaque bit du
40 plan du groupe de processus est la même que pour cette autre table. La

table du système, qui se trouve dans la table des communications (COM TAB), a une dimension fixée lors de la création du système tandis que le plan qui se trouve dans chaque segment de commande de groupe de processus a une longueur, fixée au moment du chargement initial du système, qui peut
5 être plus petite.

Dans la seconde phase, l'espace est attribué pour le compte d'un segment appartenant à un groupe de processus donné. L'espace est séquentiellement attribué dans le bloc en cours de remplissage pour ce groupe de processus (l'adresse et le volume d'espace disponible sont trouvés dans le
10 segment de commande du groupe de processus, comme expliqué ci-dessus). Lorsque l'espace disponible est insuffisant, un nouveau bloc est attribué pour le compte du groupe du processus (phase 1).

La stratégie d'attribution pour la partie 2 est similaire à celle utilisée pour la partie 3 (c'est-à-dire qu'elle comporte deux phases) mais comme
15 la première phase n'est pas exécutée pour le compte d'un groupe de processus (mais au contraire au niveau du système) les divers paramètres sont situés, sur une base centralisée, dans la table des communications. En outre, il n'est pas besoin d'un plan binaire pour la partie 2 du fait que l'espace attribué n'est jamais libéré jusqu'à une récréation du système et que l'es-
20 pace de mémoire peut être ainsi attribué de façon continue.

Les règles qui sont suivies pour réaliser l'attribution effective sont les mêmes dans les deux parties et leur liste est donnée ci-après.

a) La demande est définie par deux entités: une dimension et une dimension maximale.

25 b) Si l'espace disponible dans le bloc en cours est supérieur ou égal à la dimension maximale, il est attribué dans le bloc en cours et la dimension disponible est mise à jour (phase 2). Sinon un nouveau bloc est attribué (phase 1).

c) Si la dimension maximale est supérieure à la capacité d'un bloc, la totalité du bloc est attribuée pour satisfaire la demande; si cette dimension
30 maximale est inférieure à la capacité du bloc, la dimension maximale est attribuée et la dimension disponible est rendue égale à la capacité du bloc moins la dimension maximale.

35 6 - 4 - 1 - 2 - Détails du traitement (Attribution de la mémoire auxiliaire).

(1) Le type de la demande est vérifié pour déterminer si la fonction d'attribution est demandée pour la partie de mise à jour dynamique (partie 2) ou pour la partie du fichier de la mémoire auxiliaire réservée pour les
échanges de segments (partie 3).

40 (2) Si la demande concerne la partie 3, deux zones PGBSAS

(état d'attribution de la mémoire auxiliaire pour le groupe de processus) et PGBSADR (adresse du disque en cours dans la mémoire du groupe du processus) sont extraites dans la partie de la sous-zone du segment de commande (PGCS) du groupe de processus pour lequel l'attribution doit être effectuée.

- 5 Le numéro J de ce groupe de processus est trouvé dans le registre général d'entrée GRJPS.

(3) Si la demande concerne la partie 2, deux zones, la zone BSDUAS (état d'attribution de la partie 2 de la mémoire auxiliaire) et la zone BSDUADR (adresse du disque en cours de la mémoire auxiliaire pour la par-

- 10 tie 2) sont extraites de la table des communications. La zone BSDUAS contient à la fois le nombre de pistes disponibles restant dans le bloc en cours de remplissage et le nombre de multiplats disponibles restant dans la première de ces pistes. La zone BSDUADR contient l'adresse relative sur le disque (C, H, R) de l'enregistrement suivant à écrire dans le bloc en cours de
15 remplissage.

(4) La variable, appelée T, qui contiendra le nombre total des pistes attribuées (total signifie que soit la première, soit la dernière piste, soit encore ces deux pistes peuvent être partiellement attribuées) est mise à l'état initial "1 piste" (une piste au minimum est nécessaire).

- 20 (5) La dimension maximale demandée (GRMSZ) est comparée au nombre de multiplats disponibles restant sur la piste en cours de remplissage pour déterminer si cette dimension peut être contenue dans cette piste.

(6) Si le nombre de multiplats disponibles est égal à la dimension demandée, le nombre des pistes disponibles est diminué d'une unité.

- 25 (7) S'il n'y a pas de piste restante dans le bloc en cours de remplissage, le nombre de multiplats disponibles est remis à zéro mais s'il reste encore une piste le nombre de multiplats disponibles est mis à la pleine capacité d'une piste (compteur BSNBPT qui se trouve dans la table des communications).

- 30 (8) La dimension demandée étant plus petite que l'espace restant dans la piste en cours de remplissage, le nombre de multiplats est mis à jour de sorte qu'il donne à tout moment, la limite supérieure de la dimension de l'enregistrement suivant à écrire.

- (9) Le nombre de multiplats dans la première piste attribuée et le
35 nombre de multiplats dans la dernière piste attribuée sont chargés avec la dimension maximale demandée.

(10) Il est alors déterminé s'il reste de l'espace disponible dans le bloc en cours de remplissage.

- (11) La valeur de la dimension maximale qui dépasse le nombre de
40 multiplats disponibles restant sur la piste en cours de remplissage est

divisée par la capacité d'une piste afin de déterminer combien de pistes sont nécessaires en plus de la première.

(12) Le nombre de pistes attribuées est ajusté. Si une piste supplémentaire est nécessaire elle sera partiellement remplie.

5 (13) Si le nombre de pistes nécessaires dépasse le nombre de pistes disponibles restant dans le bloc en cours de remplissage, un nouveau bloc est attribué pour le compte du demandeur.

(14) Si le nombre de pistes disponibles satisfait la demande, le nombre est alors diminué du nombre de pistes attribuées.

10 (15) Le nombre de multiplats disponibles (BAV) est mis en mémoire dans le nombre de multiplats attribués sur la première piste (3) qui sera retourné au demandeur.

(16) Une vérification est effectuée pour déterminer si l'espace attribué couvre une partie ou la totalité de la dernière piste attribuée.

15 (17) S'il n'est besoin que d'une partie seulement de la dernière piste, le nombre de multiplats disponibles restant sur la dernière piste est mis à jour en fonction de l'espace qui vient d'être attribué sur cette piste.

(18) S'il est besoin de la totalité de la dernière piste, le nombre des multiplats restant sur la piste suivante à attribuer est 0 ou la capacité totale de la piste, selon le nombre de pistes disponibles restantes (0 ou $\neq 0$).

20 (19) Etant donné que la dimension demandée a été accordée dans le bloc en cause, le registre GRSZ contenant la dimension mise à jour (dimension qui reste à attribuer) est remis à zéro.

(20) Les autres registres de sortie sont chargés avec l'adresse relative sur le disque et la dimension sur le disque de la zone qui vient d'être attribuée.

30 (21) L'adresse relative sur le disque de l'enregistrement suivant à écrire dans le bloc en cours est alors mise à jour. S'il ne reste aucun multiplat disponible, l'adresse du disque devient sans signification et un nouveau bloc devra être attribué.

(22) Il est déterminé si l'espace restant commence à une nouvelle limite de piste.

35 (23) Si le nombre total des pistes attribuées est 1 (ce qui signifie que l'enregistrement suivant sera écrit sur une piste unique) le numéro d'enregistrement est accru d'une unité.

(24) Si plusieurs pistes ont été attribuées, et que le nombre de multiplats disponibles (BAV) est différent de zéro, étant donné que la fin du segment pour lequel l'espace a été attribué sera enregistré sur l'enregistrement n° 1 de la dernière piste attribuée, l'enregistrement suivant portera le n° 2.

40 Le numéro de lecture est mis à jour en fonction du nombre de pistes

attribuées.

(25) L'adresse du disque est remise à une limite de piste.

(26) Il est à nouveau déterminé si la demande a été présentée pour la partie 2 de la mémoire auxiliaire (mise à jour dynamique) ou pour la partie 3 (échange). Si la demande concerne la partie 3, les zones mises à jour sont mises en mémoire dans la sous-zone relative à la mémoire auxiliaire du segment de commande du groupe de processus (PGCS). Le numéro J du propriétaire du segment PGCS est trouvé dans le registre d'entrée GRJSP. Si la demande concerne la partie 2, les zones mises à jour sont mises en mémoire dans la table des communications et un code de retour de réussite est chargé dans le registre GRRC.

(27) Si la demande est pour la partie 2 mais qu'elle s'achève par une fin anormale, un code de retour est renvoyé indiquant qu'il n'y a plus de bloc disponible qui puisse être réservé pour la partie 2.

(28) La phase d'attribution de blocs est entrée en commençant par une vérification du nombre de blocs disponibles pour déterminer s'il existe encore un espace suffisant réservée pour la mise à jour dynamique.

(29) L'information BS2NEXTH (numéro de la tête du début du bloc suivant) est comparée à la capacité du cylindre pour déterminer si un nouveau cylindre doit être attribué.

(30) Le numéro de tête en cours du nouveau bloc est mis à l'état initial.

(31) Le nouveau bloc est attribué sur le cylindre suivant (le numéro de tête en cours est alors 0).

(32) Le numéro de tête du début du bloc suivant est mis à jour en fonction de la capacité de chaque bloc.

(33) Le premier enregistrement doit être écrit sur le nouveau bloc et le nombre des blocs disponibles restant est diminué d'une unité.

(34) Si l'attribution doit être effectuée pour la partie 3, le nombre de blocs disponibles est comparé à son seuil minimal. S'il est égal à ce seuil, l'état de saturation est atteint et l'activité avec la mémoire auxiliaire doit être ralentie, s'il n'est pas égale à ce seuil, l'attribution est effectuée (sous réserve qu'il reste au moins un bloc).

(35) Le bit de droite, correspondant au bloc qui vient d'être attribué, est remis à zéro dans le segment de commande (PGCS) du groupe de processus pour le compte duquel l'attribution est faite.

(36) Le compteur de blocs attribué est accru d'une unité au niveau des segments de commande de groupe de processus PGCS tandis que le compte du compteur des blocs disponibles restants est diminué d'une unité au niveau du système (dans la table des communications).

(37) Il est déterminé si la dimension demandée est plus petite que la capacité du bloc.

5 (38) Si la dimension maximale dépasse la capacité du bloc, la totalité du bloc est attribuée pour satisfaire la demande. (Par convention étant donné que la dimension demandée peut être contenue à l'intérieur du bloc le paramètre de sortie "DIM" est remis à zéro).

10 (39) Les paramètres caractérisant l'état d'attribution du bloc qui vient d'être attribué sont initialisés à la capacité maximale d'un bloc et la phase normale d'attribution d'espace à l'intérieur du bloc en cours est ré-entrée.

(40) La dimension demandée est mise à jour. (Elle est diminuée de la longueur de la zone qui vient d'être attribuée. De cette manière les paramètres sont mis à jour pour permettre au demandeur d'émettre une nouvelle demande).

15 (41) Etant donné que le bloc en cours est épuisé, les informations "multiplets disponibles" (BAV) et "pistes disponibles" (TAV) sont remises à zéro.

(42) La dimension maximale demandée est diminuée de l'espace de mémoire auxiliaire qui a été attribué.

20 (43) L'indicateur de mémoire auxiliaire (BSSW) indique si la "mise en route des groupes de processus" (PGI) a été notifiée de ralentir ses opérations du fait du manque de mémoire auxiliaire. Lorsque cet indicateur est à l'état 1, cela signifie que la notification a déjà été transmise. S'il n'a pas encore été mis à l'état 1, l'indicateur est mis alors et une notification est transmise à la "mise en route des groupes de processus" pour qu'elle arrête toute nouvelle programmation de nouveaux groupes de processus jusqu'à nouvelle notification. (Celle-ci se produit à la fin d'un groupe de processus, lorsque la mémoire auxiliaire est libérée).

30 (44) Si le nombre de blocs disponibles est tombé au-dessous du seuil minimal, l'attribution est reprise sous réserve qu'il reste au moins un bloc libre. Sinon, le processus en cours attend jusqu'à ce que des blocs soient rendus par d'autres utilisateurs. Le compteur BSWPRCT contient le nombre de processus contenus dans la file d'attente du sémaphore en attente d'un bloc libre.

35 (45) Avant que l'opération P (P-op) soit effectuée sur le sémaphore SYSBKSAT, la commande exclusive de la base des données des segments est libérée. (Cependant, le processus reste dans la section critique). Lorsqu'il est réactivé, la commande exclusive de la base des données des segments est reprise et les opérations sont reprises.

40 6 - 4 - 2 - Libération de la mémoire auxiliaire (RBSM)

6 - 4 - 2 - 1 - Description générale

Cette fonction s'applique exclusivement à la partie 3 de la mémoire auxiliaire. Elle est demandée pour le compte d'un groupe de processus dont l'identification est transmise en tant que paramètre d'entrée. Son objet est de

5 rendre tous les blocs de la mémoire auxiliaire précédemment attribués à ce groupe de processus au pool commun.

Cette fonction est nécessaire lorsqu'un groupe de processus prend fin de sorte que les diverses zones qu'il occupait dans le fichier de la mémoire auxiliaire puissent être réutilisées par d'autres utilisateurs. Les zones
10 sont mises à jour dans la table du système qui décrit le pool commun.

La partie 3 de la mémoire est divisée en blocs d'égale longueur (la dimension commune étant un paramètre créé au moment de la création du système). Chaque bloc est la plus petite quantité de mémoire qui puisse être attribuée à un groupe de processus et les blocs sont attribués un par un

15 Lorsqu'un nouveau bloc est attribué à un demandeur, ce bloc est indiqué comme occupé (au niveau du groupe de processus) dans la sous-zone relative à la mémoire auxiliaire du segment de commande du groupe de processus (PGCS) et (au niveau du système) dans la table des communications.

Ces deux tables comportent un plan binaire à l'intérieur duquel une
20 correspondance bit à bit est établie avec les blocs de mémoire physique du fichier de la mémoire auxiliaire.

Des blocs supplémentaires peuvent être attribués dynamiquement aux groupes de processus mais aucune libération de bloc n'est effectuée avant la fin du groupe de processus. A ce moment, l'attribution des blocs attribués
25 est retirée en groupe. Etant donné que le plan contenu dans le segment de commande du groupe de processus PGCS représente les blocs attribués au groupe de processus en cause, la table des communications peut être mise à jour mot par mot, en exécutant une opération logique OU exclusif entre les deux tables et la fonction de libération est ainsi exécutée très rapidement.

30 Une fois que les structures de commande sont mises à jour, comme décrit ci-dessus, et du fait qu'un grand nombre de blocs est maintenant libéré, une vérification est effectuée pour déterminer si l'indicateur du système BSSWITCH avait été précédemment mis pour indiquer que le système était entré dans l'état de saturation.

35 Dans ce cas, les demandeurs en attente s'il en existe, (processus dont les demandes ne pouvaient pas auparavant être satisfaites), sont maintenant réactivés et une notification est transmise à la "mise en route des groupes de processus" (sous réserve que le nombre des blocs disponibles soit supérieur à un seuil préalablement défini (BSMAXU)).

40 6 - 4 - 2 - 1 - Détails du traitement

1) Le plan appartenant à la table des communications est mis à jour, mot par mot, avec le contenu du plan du segment de commande du groupe de processus (PGCS).

2) Le nombre des blocs disponibles est accru du nombre des blocs
5 qui étaient occupés par le groupe de processus en cause.

3) Une vérification est effectuée pour déterminer si les activités de mise en route de groupes de processus avaient été retardées par suite d'un manque d'espace dans la mémoire auxiliaire. Le seuil BSMAXU est le seuil au-delà duquel les activités de mise en route de processus peuvent reprendre. Ainsi, le nombre en cours de blocs disponibles est comparé au seuil
10 BSMAXU + BSWPRCT (nombre de processus en attente de la mémoire auxiliaire) (en gardant à l'esprit le fait que chaque processus n'est en attente de l'attribution que d'un seul bloc).

4) Les activités de mise en route de groupes de processus (PGI) sont
15 redémarrées et l'indicateur correspondant est remis à zéro.

5) Chaque processus qu'il trouve dans la file d'attente du sémaphore SYSBKSAT est maintenant réactivé de sorte que ces processus peuvent entrer en concurrence pour l'attribution des blocs libérés.

6 - 4 - 3 - Mouvement de segment à la mémoire principale (MVMAIN)
20 et 6 - 4 - 4 - Mouvement de segment à la mémoire auxiliaire (MVBACK)

La fonction de ces deux primitives est de réaliser le transfert effectif des segments entre la mémoire auxiliaire et la mémoire principale.

Plus spécifiquement, la fonction MVMAIN effectue le transfert des segments de la mémoire auxiliaire à la mémoire principale conformément aux
25 paramètres ci-après : le contenu du registre GRJPS identifiant le nom J, P du processus pour lequel l'échange doit être effectué, le contenu du registre GRDA contenant l'adresse relative sur le disque dans le fichier de la mémoire auxiliaire (CHR) du segment à transférer, le contenu du registre GRAA contenant l'adresse absolue dans la mémoire principale de la zone
30 qui doit recevoir le segment, le contenu du registre GRSZ contenant la dimension du segment à transférer, le contenu du registre GRGPAA contenant l'adresse absolue de la zone de programme de canal attribuée et le contenu du registre GRTY qui définit le type de demande indiquant si la dimension du segment est supérieure à celle d'un unique bloc attribuable dans la
35 mémoire auxiliaire. Après achèvement du transfert, le registre GRRO contient la valeur du pointeur du bloc attribuable suivant dans la mémoire auxiliaire de sorte qu'il est possible d'accéder à ce bloc si cela est nécessaire.

Les détails de la primitive MVMAIN dépendent du caractère du
40 dispositif d'échange choisi et il n'apparaît donc nécessaire de donner ici

qu'une brève description d'une procédure typique pour effectuer un tel trans-

fert E/S. Dans son principe, la fonction de la primitive MVMAIN est de construire, dans la zone du noyau de la mémoire principale, un groupe de programmes de canaux logiques (LCP) qui comportent chacun une en-tête

- 5 de programme de canal logique, un tableau d'entrées de commande de canal et les adresses de disques ainsi que les informations de commande qui sont nécessaires pour exécuter le programme de canal. Deux formes différentes de primitives MVMAIN sont nécessaires. La première primitive MVMAIN1 est utilisée lorsque la dimension du segment à transférer à la mémoire
- 10 principale est inférieure à celle d'un bloc attribuable de sorte que le segment se trouve dans un seul bloc. La seconde primitive MVMAIN 2 est utilisée lorsque la dimension du segment est supérieure à celle d'un bloc attribuable de sorte que le segment se trouve à cheval sur plusieurs blocs (éventuellement non contigus) ce qui nécessite des commandes de canal
- 15 supplémentaires pour lire les pointeurs désignant les blocs suivants à inclure dans le transfert.

La primitive MVBACk, qui transfère le segment de la mémoire principale à la mémoire auxiliaire, est mise en oeuvre avec les paramètres ci-après transmis par les registres ci-après: le registre GRJPS qui identifie

20 le nom JP de processus pour lequel l'échange doit être effectué, le registre GRDA qui contient l'adresse relative sur le disque de la zone de la mémoire auxiliaire qui doit recevoir le segment; le registre GRAA qui contient

- gistre GRSZ qui contient la dimension du segment à transférer, le registre
- 25 GRCPAA qui contient l'adresse absolue de la zone de programme de canal attribuée, le registre GRTY qui définit le type de la demande et le registre GRDSZ utilisé dans le cas où un segment doit être écrit sur plusieurs pistes et qui contient la dimension du segment exprimée par les valeurs T (nombre de pistes attribuées), B (nombre de multipléts sur la première piste
- 30 attribuée) et B' (nombre de multipléts sur la dernière piste attribuée). Le registre GRRO est utilisé pour communiquer, comme paramètre d'entrée ou de sortie, la valeur du pointeur désignant le bloc attribuable suivant dans la mémoire auxiliaire.

6 - 5 - Exception du segment manquant

- 35 6 - 5 - 1 - Traitement des segments manquants (MISSHDL)

Avant de charger le segment manquant, le segment de pile en cours et le segment contenant le point de retour sont chargés (s'ils sont manquants) et sont rendus résidants (c'est-à-dire que l'attribut de résidence RES est mis à l'état 1 dans les en-têtes des segments). Ceci est effectué pour éviter
5 un phénomène de récurrence avec un segment manquant (un cas qui^{se} produirait à la sortie du traitement si l'un de ces deux segments était manquant). Les deux segments sont ensuite rendus échangeables (RES = 0) après que le segment manquant a été chargé.

Une fois que le segment manquant a été chargé dans la mémoire des
10 utilisateurs, la zone de programme de canal est libérée et un retour est fait à la procédure qui a déclenché l'exception, après libération de la commande exclusive du sémaphore SYS1.

6 - 5 - 1 - 2 - Détails du traitement

1) La commande exclusive des deux sémaphores du système SYSALL
15 et SYS1 est prise.

2) S'il n'y a plus de zone de programme de canal (c'est-à-dire si le compte du compteur E/S de la mémoire auxiliaire (BSIOCNT) est égal à la valeur maximale des E/S de la mémoire auxiliaire (BSIOMAX) ou à cette
20 valeur moins 1 et qu'aucune opération d'entrée/sortie n'a été demandée par la gestion des piles, la demande doit être différée. Le compte du compteur des processus en attente de la mémoire principale (MMWPRCT) est accru d'une unité et la demande est différée (après libération de la commande exclusive de la base des données des segments).

3) S'il y a au moins une zone de programme de canal (CPAREA)
25 libre, le compte E/S de la mémoire auxiliaire (BSIOCNT) est augmenté d'une unité, une entrée de zone de programme de canal (CPAREA) est choisie et l'adresse absolue de cette zone est calculée.

4) Le numéro de segment (STN, STE) de la gestion de pile en cours est extrait.

30 5) Si la pile en cours est manquante, la procédure de chargement du segment (SGLD) est appelée. Si le chargement est réussi, la pile est rendue résidante (c'est-à-dire que l'attribut RES de l'en-tête du segment de pile est mis à l'état 1). Si le chargement n'est pas réussi (dans le cas d'un descripteur de segment ayant le format fantôme, vacant ou réservé
35 ou dans le cas de saturation de la mémoire des utilisateurs) le processus est avorté.

6) Si la pile est déjà dans la mémoire des utilisateurs, elle est rendue résidante.

40 7) Le numéro de segment (STN, STE) du segment contenant le point de retour est extrait et la même procédure est exécutée que pour le

segment de pile.

8) Le numéro de segment (STN, STE) du segment manquant est extrait (à partir du message émis par le matériel).

9) La procédure de chargement de segment (SGLD) est appelée et si le chargement n'est pas réussi le processus est avorté. Si le chargement réussit, le retour est transmis au processus qui a provoqué la procédure entraînant l'exception. Dans les deux cas, les bits de résidence (RES) sont remis à zéro dans les en-têtes des segments, l'attribution de zone de programme de canal est retirée (aux étapes 10, 12 ci-après) et une sortie est faite, après libération de la commande exclusive du sémaphore du système SYS1.

10) La zone de programme de canal (CPAREA) est libérée, le bit correspondant dans le plan des zones de programme de canal est remis à zéro et le compte du compteur des opérations E/S de la mémoire auxiliaire est diminué d'une unité.

11) Si la mémoire des utilisateurs n'est pas saturée à ce moment (indicateur IOSTOP = 0) et s'il y a un ou plusieurs processus en attente sur le sémaphore SYSMMAT (pas de zone de programme de canal libre), le compte du compteur des processus en attente de la mémoire principale (MMWPRCT) est diminuée d'une unité et une opération V-op est effectuée sur ce sémaphore pour indiquer qu'une zone de programme de canal (CPAREA) a été libérée.

12) Si la mémoire des utilisateurs est saturée (IOSTOP = 1) et si la nouvelle valeur du compteur BSIOCNT est zéro, tous les segments de la mémoire des utilisateurs sont translatables et la dernière phase de la primitive AUM (échange général et compression de la mémoire) peut être reprise. Ensuite, le processus qui avait une demande d'attribution de mémoire des utilisateurs non satisfaite par suite de la saturation est réactivé.

TABLES DES MATIERES

	<u>1 - Introduction</u>
	<u>2 - Matériel de traitement</u>
	<u>3 - Structures d'informations visibles pour le matériel</u>
5	3 - 1-Introduction à l'adressage segmenté
	3-2 - Organisation des commandes et du système d'exploitation dans la mémoire principale.
	<u>4 - Structures d'informations visibles pour l'utilisateur</u>
	4- 1 -Classification des procédures et processus des utilisateurs
10	4-2 - Structures adressées à partir de la base du système
	4-3 - Adressage à partir des blocs de commande de groupe de processus
	4-4 - Protection Inter-processus
	<u>5 - Gestion de la mémoire virtuelle</u>
	5-1 - Description générale
15	5-1-1 - Gestion des segments
	5-1-2- Gestion de la mémoire principale
	5-1-3- Gestion de la mémoire auxiliaire
	5-1-4- Traitement des segments manquants
	5-1-5 - Gestion des piles
20	5-1-6 - Services de la mémoire virtuelle
	5-2 - Structures des procédures: Complexes, composantes, Primitives
	5-3- Interfaces entre les structures des procédures
	5-4 - Structures de données
	5-4-1- Registres généraux
25	5-4-2- Table des communications
	5-4-3- En-têtes de segment et de trous
	5-4-4- Descripteurs de segments
	5-4-5- Segments de commande de groupe de processus
	5-4-6- Table d'activité des travaux
30	5-4-7- Base du système du matériel
	5-4-8- Sémaphores du système
	5-4-9- Résumé: Structure des données
	<u>6 - Description dynamique des primitives</u>
	6-1 - Résumé
35	6-2- Gestion de la mémoire principale
	6-2- 1- Attribution de la mémoire des utilisateurs (AUM)
	6-2-2- Changement d'attribution de la mémoire des utilisateurs (CUM)
	6-2-3- Attribution de la mémoire de commande (ACM)
	6-2-4- Libération de la mémoire principale (RMM)
40	6-3- Gestion des segments

- 6-3-1- Changement des segments (SGLD)
- 6-3-2- Déchargement des segments (SGUNLD)
- 6-3-3- Changement de la dimension de segments (SGSIZE)
- 6-3-4- Modification d'attributs des segments (SGATTR)
- 5 6-3-5- Mise du bit de disponibilité (SDAVL)
- 6-3-6- Restauration du bit de disponibilité (SDUNAL)
- 6-3-7- Libération de la mémoire virtuelle par un groupe de processus (RELVM)
- 6-3-8- Création de descripteur de segment indirect -(ISDCRT)
- 10 6-3-9- Suppression du descripteur de segment indirect (ISDDEL)
- 6-3-10- Mouvement de segment (SGMOVE)
- 6-3-11- Déchargement de segment spécial (LDUNLD)
- 6-3-12- Autres fonctions
- 6-4- Gestion de la mémoire auxiliaire
- 15 6-4-1- Attribution de la mémoire auxiliaire (ABSM)
- 6-4-2- Libération de la mémoire auxiliaire (RBSM)
- 6-4-3- Mouvement de segment à la mémoire principale (MVMAIN)
- 6-4-4- Mouvement de segment à la mémoire auxiliaire (MVBACK)
- 6-5- Exception de segment manquant
- 20 6-5-1- Traitement des segments manquants (MISSHDL)

Demandes connexes

- "Buffer store" invention de CURLEY , DONAHUE , MARTLAND , FRANKLIN , déposée aux U.S.A le 5 Octobre 1972 sous le N° 295.301 par la Société Honeywell Information Systems domiciliée aux U.S.A .
- 5 - "Variable Masking for segmented Memory" invention de MARTLAND CURLEY déposée aux U.S.A le 5 Octobre 1972 sous le N°295 303 par la Société Honeywell Information Systems domiciliée aux U.S.A.
- "Override Hardware for Main Store Sequencer" , invention de DONAHUE , déposée aux U.S.A le 5 Octobre 1972 sous le N° 295.418 par
- 10 la Société Honeywell Information Systems domiciliée aux U.S.A .
- "Main Memory Sequencer" invention de DONAHUE , CURLEY , FRANKLIN , MARTLAND , CORNARO déposée le 5 octobre 1972 aux U.S.A sous le N° 295.331 par la Société Honeywell Information Systems domiciliée aux U.S.A .
- 15 -"Main memory reconfiguration" invention de CURLEY , FRANKLIN, MARTLAND , DONAHUE , CORNARO déposée aux U.S.A le 5 Octobre 1972 sous le N° 295.417 par la Société Honeywell Information Systems domiciliée aux U.S.A .
- "Protection des informations dans un ordinateur multiprogrammable/ multitraitement " , invention de BRADLEY , FRANKLIN, APPELL , BIENVENU , LEPICARD , DE RIVET déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N°7342706.
- 20 -"Appels de procédures et Opérations de piles" , invention de BRADLEY FRANKLIN, APPELL , BOGAERT, MASSUART déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 7342705.
- 25 - "Synchronisation des processus utilisant des sémaphores" , invention de BIENVENU, CARRE, VERDIER, déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 7342691
- 30 -"Développements d'adresse segmentée" , invention de BIENVENU , déposée en France le 16 Mai 1973 par la Compagnie demanderesse sous le N° 73 17765 .
- "Instructions P & V sur sémaphores pour synchronisation des Processus " , invention de BIENVENU , CARRE , LUU , VERDIER déposée en
- 35 France simultanément à la présente demande par la Compagnie demanderesse sous le N° 7342697.
- "Bloc de contrôle de processus" , invention de BIENVENU, CARRE LUU VERDIER déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 7342689.
- 40 -"Base du système" , invention de CARRE , LUU, DE RIVET, VERDIER

déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 73 42698.

- 5 -"Système de gestion des processus pour un processeur central" invention de CARRE, LUU, DE RIVET, VERDIER déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 73 42694.

-"Gestionnaire de mémoire virtuelle" invention de CARRE, HIOT, MICHEL, ROLLAND, déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 73 42707.

- 10 -"Protection de l'information dans un système de multraitement de l'information mettant en oeuvre un concept d'anneaux pour représenter les différents niveaux de privilèges entre processus", invention de BRADLEY, FRANKLIN, APPELL, LEPICARD, DE RIVET déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 73 42706.

-"Instructions de gestion de processus pour processeur de données" invention de CARRE, LUU, DE RIVET, VERDIER déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 73 42692.

- 20 -"Instructions d'initialisation de sémaphores pour synchronisation des Processus" invention de BRADLEY, FRANKLIN, LUU, DE RIVET, déposée en France simultanément à la présente demande par la Compagnie demanderesse sous le N° 73 42696.

REVENDEICATIONS

- 1- Ordinateur comportant un système d'emmagasinement d'informations comprenant en combinaison une mémoire principale à accès sélectif et une mémoire de masse auxiliaire, les informations dans ledit système d'emmaga-
- 5 sinage étant divisées en segments de dimensions variables, l'adresse physique dans le système d'emmagasinement de chaque segment étant enregistrée dans un descripteur de segment correspondant mis en mémoire dans un emplacement identifié par l'adresse logique de chaque segment caractérisé en ce qu'il comporte un système pour attribuer à un segment en attente, enregistré
- 10 à ce moment dans la mémoire auxiliaire, l'espace demandé dans la mémoire principale ledit système comportant:
- des moyens pour mettre en mémoire à un premier emplacement de mémoire la dimension de la zone d'adresses vides la plus grande actuellement présente dans la mémoire principale;
 - 15 - des moyens pour comparer ladite dimension de zone d'adresses vides la plus grande avec la dimension dudit espace demandé et, si l'espace demandé est plus petit que ladite dimension la plus grande, pour comparer séquentiellement la dimension de chaque zone vide de la mémoire principale à la dimension dudit espace demandé jusqu'à ce que la première zone vide de plus grande dimension soit
 - 20 localisée;
 - des moyens pour attribuer un espace d'adresses dans ladite première zone vide audit segment en attente et;
 - des moyens pour déterminer et mettre en mémoire audit premier
 - 25 emplacement de mémoire la dimension de la nouvelle zone d'adresses vides la plus grande dans la cas où l'espace a été attribué dans la zone d'adresses vides qui était antérieurement la plus grande.
- 2 - Ordinateur selon la revendication 1 caractérisé en ce que ledit descripteur de segment contient une zone de dimension qui indique l'étendue
- 30 d'espace d'adresses de mémoire qui est occupé actuellement par le segment correspondant, lesdits moyens prévus pour comparer ladite zone d'adresses vides la plus grande à la dimension dudit espace demandé comportant:
- des moyens pour extraire le descripteur du segment en attente et;
 - des moyens pour comparer la zone de dimension contenue dans le-
 - 35 dit descripteur dudit segment en attente à la dimension de la zone d'adresses vides la plus grande mise en mémoire audit premier emplacement de mémoire.
- 3- Ordinateur selon l'une quelconque des revendications 1 et 2 caractérisé en ce que chacun des descripteurs de segment comporte une zone
- 40 d'état indiquant si le segment correspondant est ou non actuellement

emmagasiné dans la mémoire principale :

- lesdits moyens prévus pour comparer la dimension de ladite zone d'adresses vides la plus grande avec ledit espace demandé étant automatiquement actionné chaque fois que la zone d'état d'un

5

descripteur de segment correspondant à un segment contenant des informations nécessaires pour le traitement indique que le segment nécessaire n'est pas présent dans la mémoire principale.

4 - Ordinateur selon l'une quelconque des revendications 1 à 3 caractérisé en ce que lesdits moyens prévus pour comparer séquentiellement la dimension de chaque zone vide de la mémoire principale à la dimension dudit

10

espace demandé comprennent :

- des moyens pour mettre en mémoire, dans un emplacement adjacent à chacune des zones vides de la mémoire principale, un lien contenant l'adresse physique du lien adjacent à la zone vide suivante,

15

dans l'ordre de changement des adresses physiques de la mémoire principale;

- des moyens pour mettre en mémoire, dans un second emplacement de mémoire, l'adresse du lien de la première zone d'adresses vides dans la séquence;

20

- des moyens pour mettre en mémoire, dans un troisième emplacement de mémoire l'adresse du lien adjacent à la dernière zone vide de la séquence.

5 - Ordinateur selon la revendication 4 caractérisé en ce qu'il comporte des moyens pour modifier les liens associés aux zones vides adjacentes à une zone vide dont l'espace d'adresses est entièrement attribué audit segment en attente.

25

6 - Ordinateur selon l'une quelconque des revendications 1 à 5 caractérisé en ce qu'il comporte des moyens pour sélectionner un segment à extraire de ladite mémoire principale dans le cas où ladite zone d'adresses vides la plus grande est plus petite que ledit espace demandé.

30

7 - Ordinateur selon la revendication 6 caractérisé en ce que lesdits moyens prévus pour sélectionner un segment à extraire de la mémoire principale comportent:

- des moyens associés à chacun desdits segments pour indiquer si le segment est éligible ou non éligible pour être extrait de la mémoire principale;

35

- des moyens pour comparer séquentiellement la dimension dudit espace demandé à la dimension de la zone combinée composée de chacun des segments éligibles pour être extraits et d'une zone vide quelconque adjacente audit segment.

40

8 - Ordinateur selon la revendication 7 caractérisé en ce qu'il comporte des moyens pour attribuer l'espace d'adresses audit segment en attente dans la première desdites zones combinées ayant une dimension supérieure à celle de l'espace demandé.

5 9 - Ordinateur selon l'une quelconque des revendications 7 et 8 caractérisé en ce qu'il comporte des moyens pour mettre en mémoire, en un emplacement adjacent à chacun desdits segments éligibles pour être extraits de la mémoire principale, un lien contenant l'adresse physique du lien adjacent au segment éligible pour être extrait immédiatement suivant, l'ordre
10 de changement d'adresses physiques dans la mémoire principale;

- des moyens pour mettre en mémoire l'adresse du lien adjacent au segment éligible dont l'adresse est la plus élevée; et

15 - des moyens pour mettre en mémoire l'adresse du lien adjacent au segment éligible dont l'adresse est la plus faible de telle sorte que la comparaison séquentielle desdits segments éligibles pour être extraits de la mémoire principale peut être effectué.

10 - Ordinateur selon la revendication 9 caractérisé en ce que ladite comparaison séquentielle est effectuée par ordre d'adresses physiques décroissantes.

20 11 - Ordinateur selon l'une quelconque des revendications 7 à 11 caractérisé en ce qu'il comporte des moyens pour mettre en mémoire la dimension de la plus grande desdites zones combinées et pour comparer la dimension dudit espace demandé à la dimension de ladite zone combinée la plus grande et n'effectuer une comparaison séquentielle que si la dimension demandée est inférieure à celle de ladite zone combinée la plus grande.
25

12 - Ordinateur selon l'une quelconque des revendications 7 à 11 caractérisé en ce que les moyens associés à chacun desdits segments pour indiquer si un segment est éligible pour être extrait de la mémoire principale sont actionnés par l'exécution d'une instruction de déchargement du segment
30 émise par l'utilisateur.

13 - Ordinateur selon l'une quelconque des revendications 6 à 12 caractérisé en ce que lesdits moyens prévus pour sélectionner un segment à extraire comportent :

35 - des moyens pour mettre en mémoire une valeur de priorité associée à chacun de tous les segments qui sont susceptibles d'être extraits de la mémoire principale; et
- des moyens pour empêcher l'extraction d'un segment donné lorsqu'il existe un segment quelconque dans la mémoire principale qui a une valeur de priorité inférieure et qui, s'il est extrait, fournit un
40 espace suffisant pour recevoir ledit segment en attente.

14 - Ordinateur selon l'une quelconque des revendications 6 à 13 caractérisé en ce qu'il comporte une zone d'état contenue dans chaque descripteur de segment pour indiquer si le segment correspondant a été modifié ou agrandi par rapport à la version du segment enregistré dans ladite mémoire auxiliaire; et

- des moyens pour transférer un segment choisi pour être extrait de la mémoire principale à ladite mémoire auxiliaire uniquement dans le cas où ledit segment a été modifié ou agrandi.

15 - Ordinateur selon l'une quelconque des revendications 1, 4, 9 et 11 caractérisé en ce que lesdits descripteurs sont contenus dans une table de segments, lesdits moyens pour mettre en mémoire les dimensions des plus grandes desdites zones et les adresses de liens étant constituées par une table des communications.

16 - Ordinateur comprenant un système d'emmagasinement d'informations, qui comporte une mémoire principale à accès sélectif et une mémoire de masse, un processeur central et une pluralité de registres de commande de processus coopérant avec ledit processeur, les informations dudit système étant cloisonnées en segments de dimensions variables caractérisé en ce qu'il comporte des moyens pour transférer automatiquement les segments entre ladite mémoire principale et ladite mémoire de masse, les dits moyens comprenant:

- au moins une première table de segments comprenant une pluralité de descripteurs de segments qui comportent les adresses de base des segments contenant des informations se rapportant à une procédure d'utilisateur;

- au moins une seconde table de segments comprenant une pluralité de descripteurs de segments qui comportent les adresses de base de segments contenant des instructions à exécuter dans le cas où des segments décrits dans la première table sont absents de ladite mémoire principale lorsqu'ils sont nécessaires au cours de l'exécution de ladite procédure d'utilisateur;

- des moyens pour mettre en mémoire des informations de commande de processus dans lesdits registres de commande de processus, lesdites informations de commande comportant des informations d'adresses désignant l'emplacement de mémoire à la fois de la première et de la deuxième table de segments de telle sorte que lesdites procédures du système et de l'utilisateur peuvent être exécutées dans un espace d'adresses commun sans modifier le contenu desdits registres de commande de processus.

17 - Ordinateur selon la revendication 16 caractérisé en ce qu'il

comporte des moyens pour empêcher que les segments contenant lesdites procédures du système soient extraits de la mémoire principale.

18 - Ordinateur selon l'une quelconque des revendications 16 et 17 caractérisé en ce qu'il comporte des moyens pour empêcher la modification
5 desdites procédures du système par l'exécution desdites procédures d'utilisateurs.

19 - Ordinateur à mémoire virtuelle comportant une mémoire principale à accès sélectif, une mémoire de masse et des moyens pour transférer automatiquement des blocs d'informations entre ladite mémoire principale et la-
10 dite mémoire de masse caractérisé en ce qu'il comporte des moyens pour détecter et supprimer les nombres excessifs de transferts d'informations entre ladite mémoire de masse et ladite mémoire principale par unité de temps lesdits moyens comprenant:

- une source de signaux d'horloge;
- 15 - des moyens pour mettre en mémoire une valeur de temps initiale fournie par ladite source lorsque se produit le premier transfert d'une série de transferts;
- un compteur;
- des moyens pour mettre un compte initial dans ledit compteur lorsque
20 se produit ledit premier transfert;
- des moyens pour accroître le compte dudit compteur chaque fois que se produit un transfert dans ladite série;
- des moyens pour comparer le signal d'horloge actuel de ladite source à ladite valeur de temps initiale pour produire une valeur de temps
25 écoulé lorsque chacun desdits transferts se produit;
- des moyens pour produire une valeur de taux de rotation si ladite valeur de temps écoulé dépasse une durée prédéterminée, ladite valeur de taux de rotation étant égale au rapport de la valeur contenue dans ledit compteur et ladite valeur de temps écoulé;
- 30 - des moyens pour comparer ladite valeur de taux de rotation à une valeur de seuil maximal prédéterminée; et
- des moyens pour décharger un ensemble choisi d'informations de ladite mémoire si ladite valeur de taux de rotation dépasse ladite valeur de seuil maximale.

35 20) Ordinateur selon la revendication 19 caractérisé en ce qu'il comporte des moyens pour remettre à l'état initial ledit compteur et ladite valeur de temps initiale à la suite de la production de chaque valeur de taux de rotation.

21 - Ordinateur selon l'une quelconque des revendications 19 et 20
40 caractérisé en ce que les informations mises en mémoire dans ladite mémoire

principale sont divisées en tâches identifiables;

- des moyens étant prévus pour mettre en mémoire pour chacune des dites tâches une valeur déclarée représentant les besoins de mémoire principale prévisibles;

5

- des moyens étant prévus pour contrôler, pour chacune des dites tâches, la valeur instantanée de l'étendue de mémoire principale qui lui est effectivement consacrée; et

- des moyens fonctionnant en réponse à un taux de rotation excessif pour décharger de la mémoire virtuelle la tâche ayant le rapport de la valeur instantanée à la valeur déclarée la plus élevée.

10

22 - Ordinateur selon l'une quelconque des revendications 19 à 21 caractérisé en ce qu'il comporte en outre des moyens pour empêcher le chargement dans la mémoire virtuelle d'informations relatives à une nouvelle tâche jusqu'à ce que ledit taux de rotation soit devenu inférieur à une valeur de seuil minimale, ladite valeur de seuil minimale étant inférieure à ladite valeur de seuil maximale.

15

23 - Ordinateur selon la revendication 21 caractérisé en ce qu'il comporte des moyens pour assurer une disponibilité de la mémoire principale pour au moins certaines tâches choisies au moins égale à ladite valeur déclarée.

20

24 - Ordinateur à mémoire virtuelle comprenant un système d'emmagasinement d'informations comportant une mémoire principale à accès sélectif et une mémoire de masse, des moyens pour charger des informations dans ledit système d'emmagasinement, lesdites informations étant cloisonnées en segments occupant des espaces d'adresses variables, un processeur coopérant avec ledit système d'emmagasinement et des moyens de traduction pour traduire les adresses logiques des informations nécessaires audit processeur en adresses physiques qui spécifient l'emplacement réel des informations demandées dans ledit système d'emmagasinement caractérisé en ce que lesdits moyens de traduction comportent un descripteur de segments associé à chacun desdits segments et comportant une adresse de mémoire principale si le segment est présent dans ladite mémoire principale et une adresse de mémoire de masse si le segment n'est pas présent dans la mémoire principale;

25

30

- un dispositif de commande pour exécuter les transferts d'informations entre ladite mémoire de masse et ladite mémoire principale conformément à des instructions contenues dans un programme de canal;

35

- des moyens pour attribuer dans ladite mémoire principale un espace qui convient pour recevoir un segment contenant des informations requises par ledit processeur mais alors absentes de

40

ladite mémoire principale;

- des moyens pour produire automatiquement un programme de canal exécutable par ledit dispositif de commande, ledit programme comportant à la fois l'adresse de l'espace attribué dans la mémoire principale et l'adresse dans la mémoire de masse du segment absent;
- des moyens pour mettre en mémoire ledit programme de canal dans la mémoire principale pour son exécution par ledit dispositif de commande afin de transférer le segment absent à la mémoire principale; et
- des moyens pour remplacer ladite adresse de mémoire de masse dans ledit descripteur de segment absent par l'adresse de l'espace attribué.

25 - Ordinateur selon la revendication 24 caractérisé en ce qu'il comporte une zone dans chaque descripteur pour empêcher ledit processeur d'accéder à un segment en cours de transfert jusqu'à ce que ledit programme de canal ait été exécuté.

26 - Ordinateur comportant une mémoire de masse comportant un espace d'adresses divisé en une pluralité de blocs d'espaces d'adresses de même dimension et des moyens pour attribuer automatiquement les espaces d'adresses dans ladite mémoire de masse à des tâches individuelles divisées en segments d'informations caractérisé en ce qu'il comporte:

- des moyens pour attribuer les blocs un par un auxdites tâches individuelles;
- des moyens pour attribuer séquentiellement dans chaque bloc des espaces d'adresses aux segments d'informations de la tâche individuelle; et
- des moyens pour attribuer un nouveau bloc à ladite tâche lorsque l'espace d'adresses dudit bloc est complètement occupé par lesdits segments.

27 - Ordinateur selon la revendication 26 caractérisé en ce qu'il comporte un répertoire général contenant des mots de répertoire, chaque mot de répertoire contenant l'adresse d'un bloc et une indication de la disponibilité ou de la non disponibilité dudit bloc:

- des moyens d'indexage indiquant le mot du dernier bloc attribué dans ledit répertoire; et
- des moyens pour explorer séquentiellement ledit répertoire à partir dudit dernier bloc attribué afin de déterminer le bloc libre suivant à attribuer.

28 - Ordinateur selon la revendication 27 caractérisé en ce qu'il

comporte un répertoire associé à chacune desdites tâches contenant des mots indiquant les adresses des blocs attribuées à ladite tâche; et

- des moyens pour comparer ledit répertoire associé à une tâche audit répertoire général lorsque ladite tâche doit être extraite de la mémoire; et

5

- des moyens fonctionnant en réponse à la comparaison pour modifier ledit répertoire général pour indiquer la disponibilité des blocs associés à ladite tâche.

29 - Ordinateur selon l'une quelconque des revendications 26 à 28

10 caractérisé en ce qu'il comporte des moyens de comptage pour compter le nombre total des blocs non attribués; et

- des moyens pour empêcher le chargement de nouvelles tâches dans ladite mémoire de masse lorsque ledit compte tombe au-dessous d'un seuil minimal.

15

30 - Ordinateur selon l'une quelconque des revendications 26 à 29 caractérisé en ce qu'il comporte des moyens de mémoire pour enregistrer l'étendue de l'espace restant disponible dans chaque bloc en cours qui a reçu la dernière information;

- des moyens associés à chacun desdits segments pour mettre en mémoire sa dimension en cours et sa dimension maximale permmissible; et
- des moyens pour réserver la totalité dudit espace restant dans ledit bloc en cours pour un segment donné à mettre en mémoire lorsque la dimension maximale dudit segment est supérieure à l'étendue dudit espace restant.

20

25

31 - Ordinateur selon la revendication 30 caractérisé en ce qu'il comporte des moyens pour comparer ladite dimension maximale permmissible à l'étendue dudit espace restant et dans le cas où ladite dimension maximale est plus petite pour soustraire ladite dimension en cours de ladite étendue d'espace restant pour donner une nouvelle mesure dudit espace restant.

30

32 - Ordinateur comportant une mémoire et des moyens pour extraire dynamiquement de ladite mémoire un segment d'information choisi parmi une pluralité de segments d'informations de dimensions variables précédemment enregistrés dans ledit dispositif de mémoire en vue de libérer un espace demandé dans ladite mémoire caractérisé en ce qu'il comporte:

35

- des moyens pour mettre en mémoire les attributs de chaque segment indiquant la dimension dudit segment et l'ordre de priorité qui lui a été attribué;
- des moyens pour lire et comparer les attributs des deux segments qui ont été enregistrés le plus anciennement dans ladite mémoire; et
- des moyens pour choisir entre les deux segments les plus anciens

40

celui qui a la plus faible valeur de priorité comme le meilleur segment candidat à être extrait.

5 33 - Ordinateur selon la revendication 32 caractérisé en ce qu'il comporte des moyens pour lire ensuite séquentiellement les attributs desdits segments dans l'ordre dans lequel lesdits segments ont été initialement enregistrés et pour comparer les attributs de chaque segment aux attributs du segment précédemment choisi comme meilleur candidat de façon à choisir à chaque fois un nouveau meilleur candidat.

10 34 - Ordinateur selon la revendication 33 caractérisé en ce qu'il comporte en outre des moyens pour choisir entre deux segments ayant la même valeur de priorité et une dimension supérieure ou égale à l'espace demandé, celui ayant la plus petite dimension.

15 35 - Ordinateur selon l'une quelconque des revendications 33 et 34 caractérisé en ce qu'il comporte en outre des moyens pour choisir entre deux segments ayant la même dimension et la même priorité celui qui a été enregistré le plus anciennement dans ladite mémoire.

20 36 - ordinateur selon l'une quelconque des revendications 33 à 35 caractérisé en ce que lesdits moyens prévus pour lire séquentiellement lesdits attributs comportent une zone de liaison mise en mémoire avec chacun desdits segments, ladite zone de liaison comportant l'adresse du segment suivant par ordre d'ancienneté.

37 - Ordinateur selon l'une quelconque des revendications 32 à 36 caractérisé en ce qu'il comporte en outre une mémoire auxiliaire dans laquelle lesdits segments existants sont renvoyés;

- 25 - une table d'informations identifiant les segments associés à chacune d'une pluralité de tâches indépendantes;
- des moyens pour mettre en mémoire pour chacune desdites tâches un nombre déclaré indiquant le nombre de segments normalement nécessaire dans ladite mémoire principale pour l'exécution efficace
- 30 de ladite tâche;
- des moyens pour contrôler le nombre instantané des segments utilisés par ladite tâche qui sont actuellement enregistrés dans la mémoire;
- 35 - des moyens pour identifier la tâche qui est actuellement en cours d'exécution par le processeur initial de l'ordinateur; et
- des moyens pour limiter les segments candidats à être extraits aux segments de ladite tâche en cours d'exécution lorsque le nombre instantané de ladite tâche est supérieur au nombre déclaré pour ladite tâche.

40 38 - Ordinateur selon la revendication 37 caractérisé en ce qu'il

comporte des moyens pour exclure lesdits segments de ladite tâche en cours d'exécution desdits segments candidats lorsque l'edit nombre instantané est inférieur au nombre déclaré.

- 5 39 - Ordinateur selon l'une quelconque des revendications 37 et 38 caractérisé en ce qu'il comporte des moyens pour identifier des tâches choisies en tant que tâches préférentielles et des moyens pour exclure desdits segments candidats les segments desdites tâches préférentielles dont le nombre instantané est inférieur au nombre déclaré.

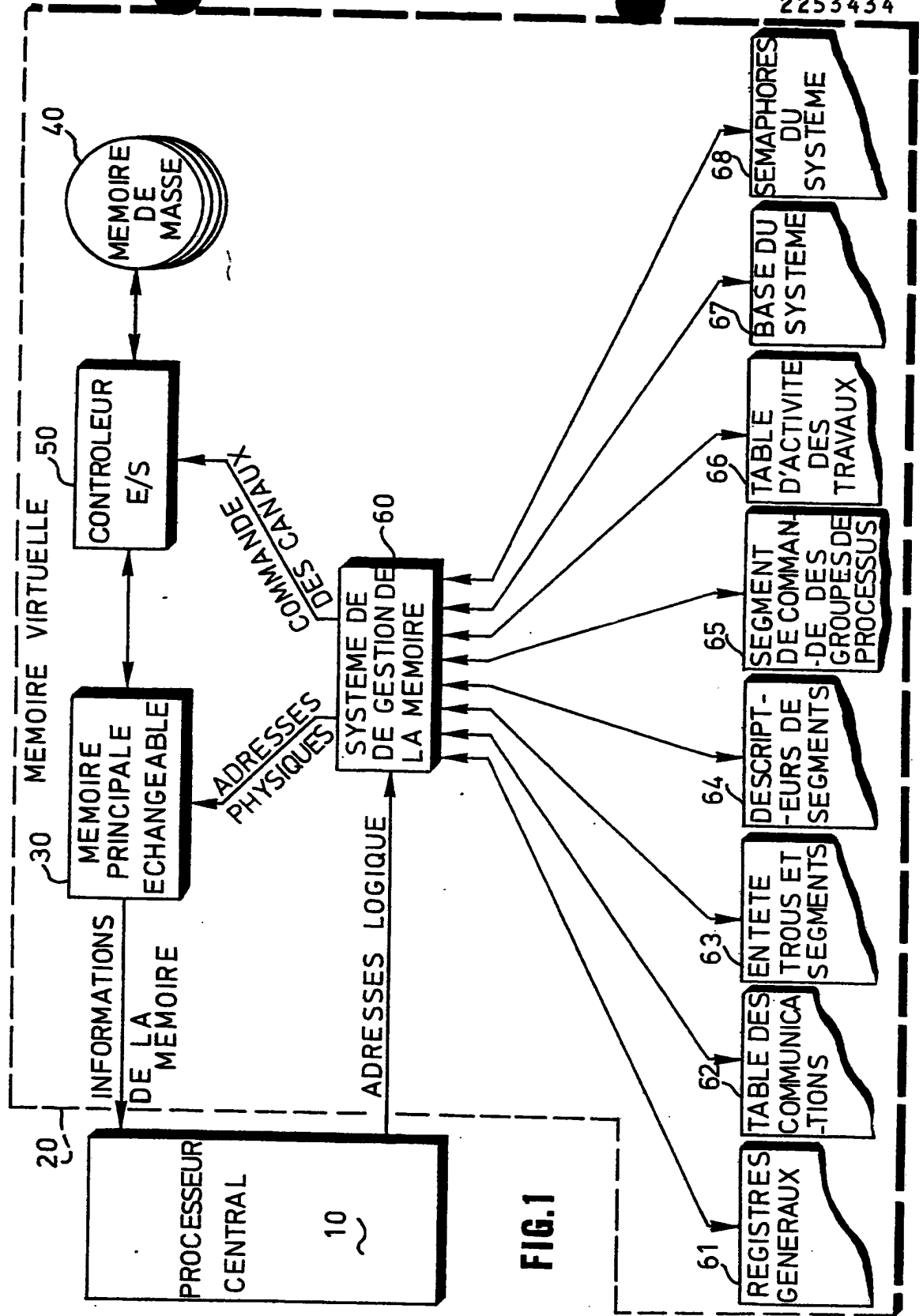


FIG.1

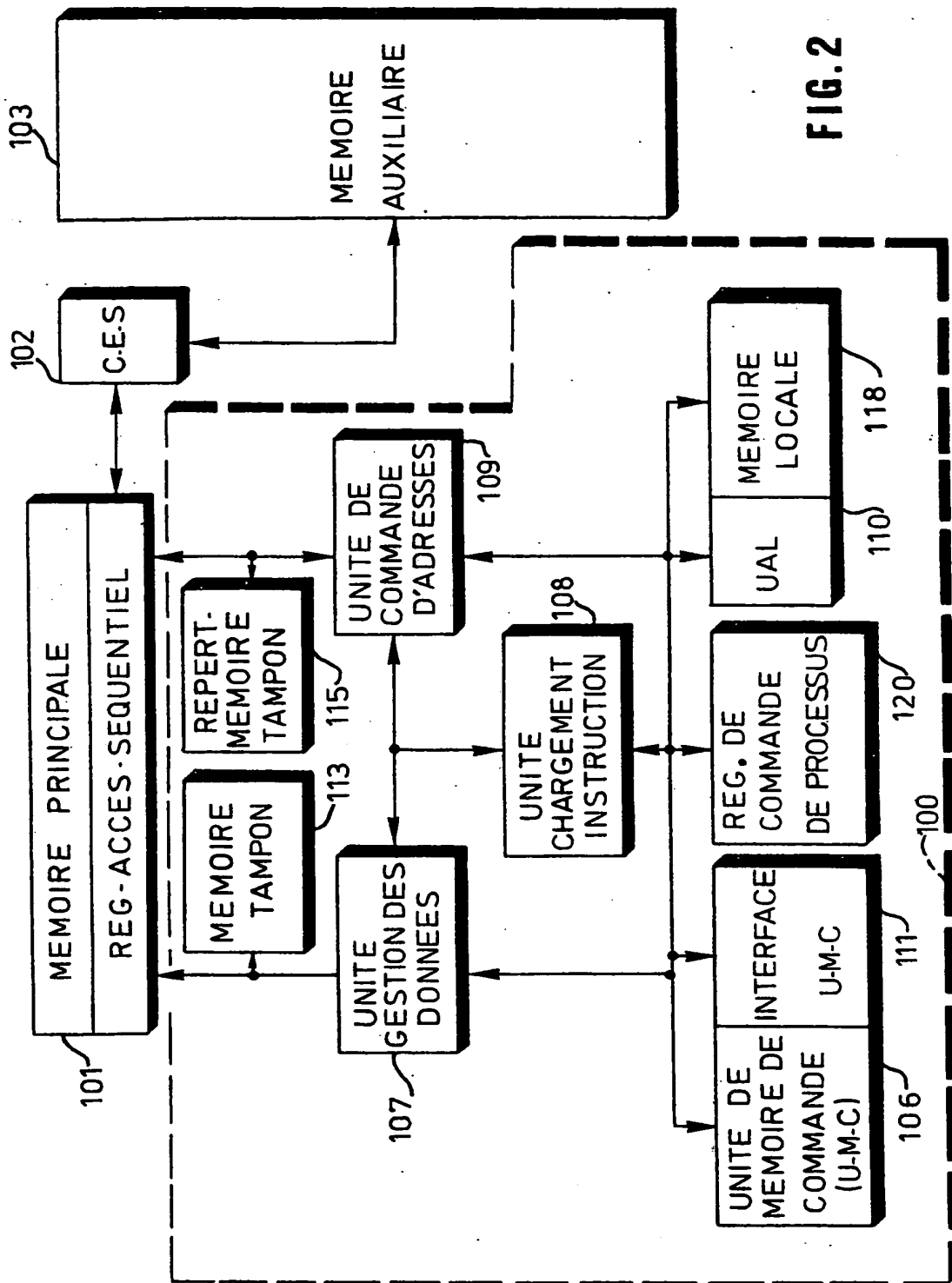


FIG. 2

FIG. 3

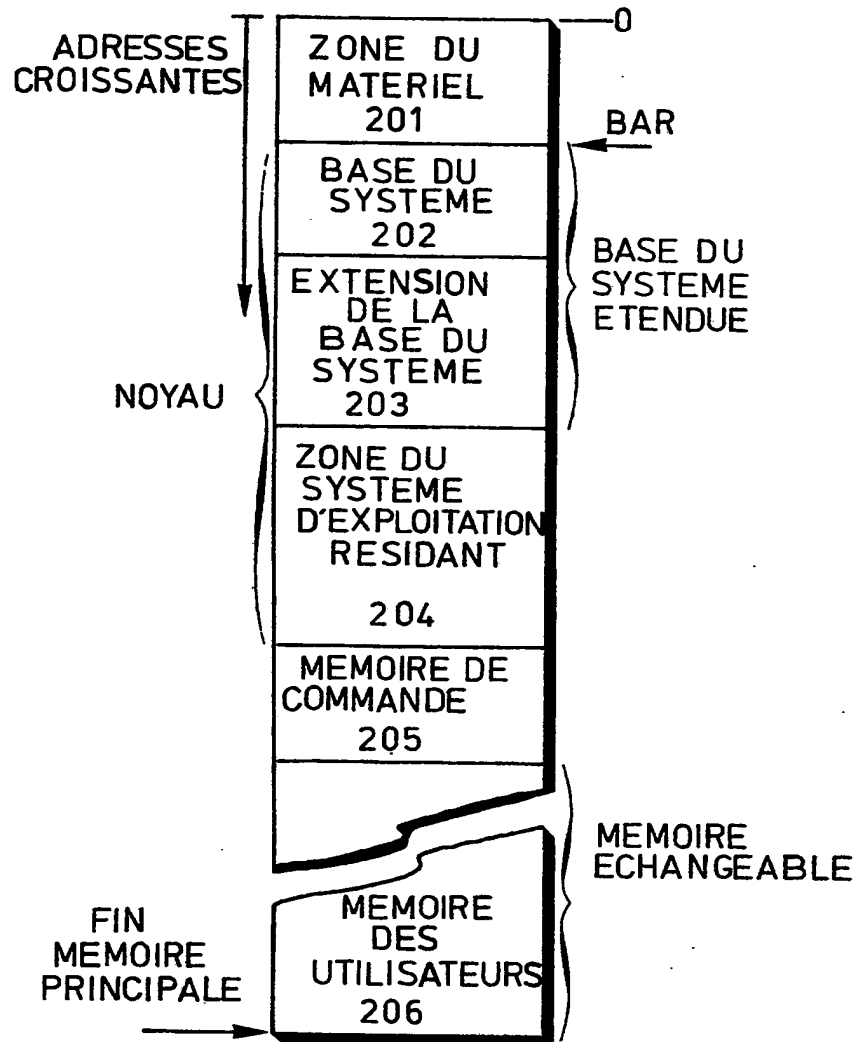
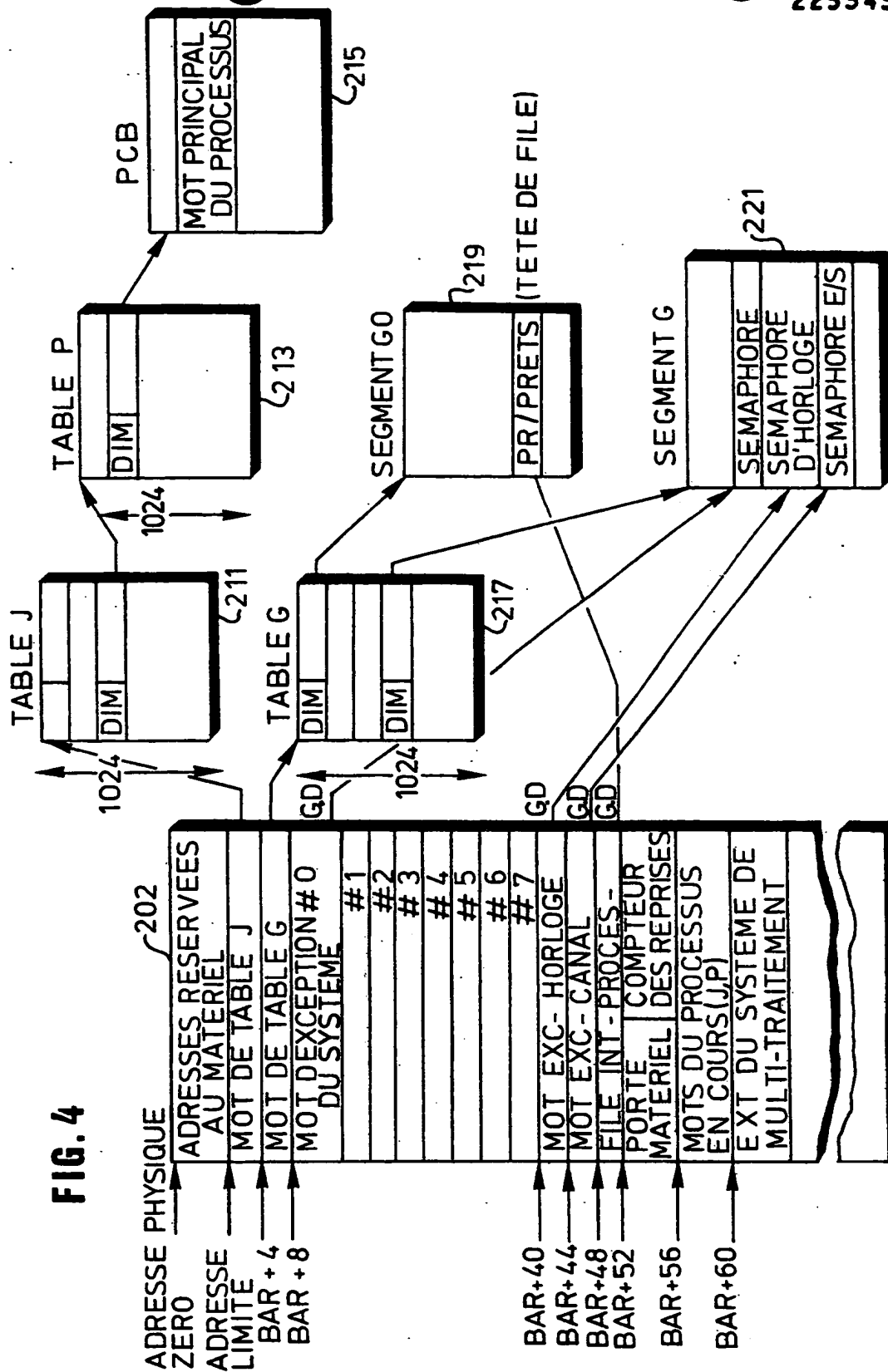


FIG. 4



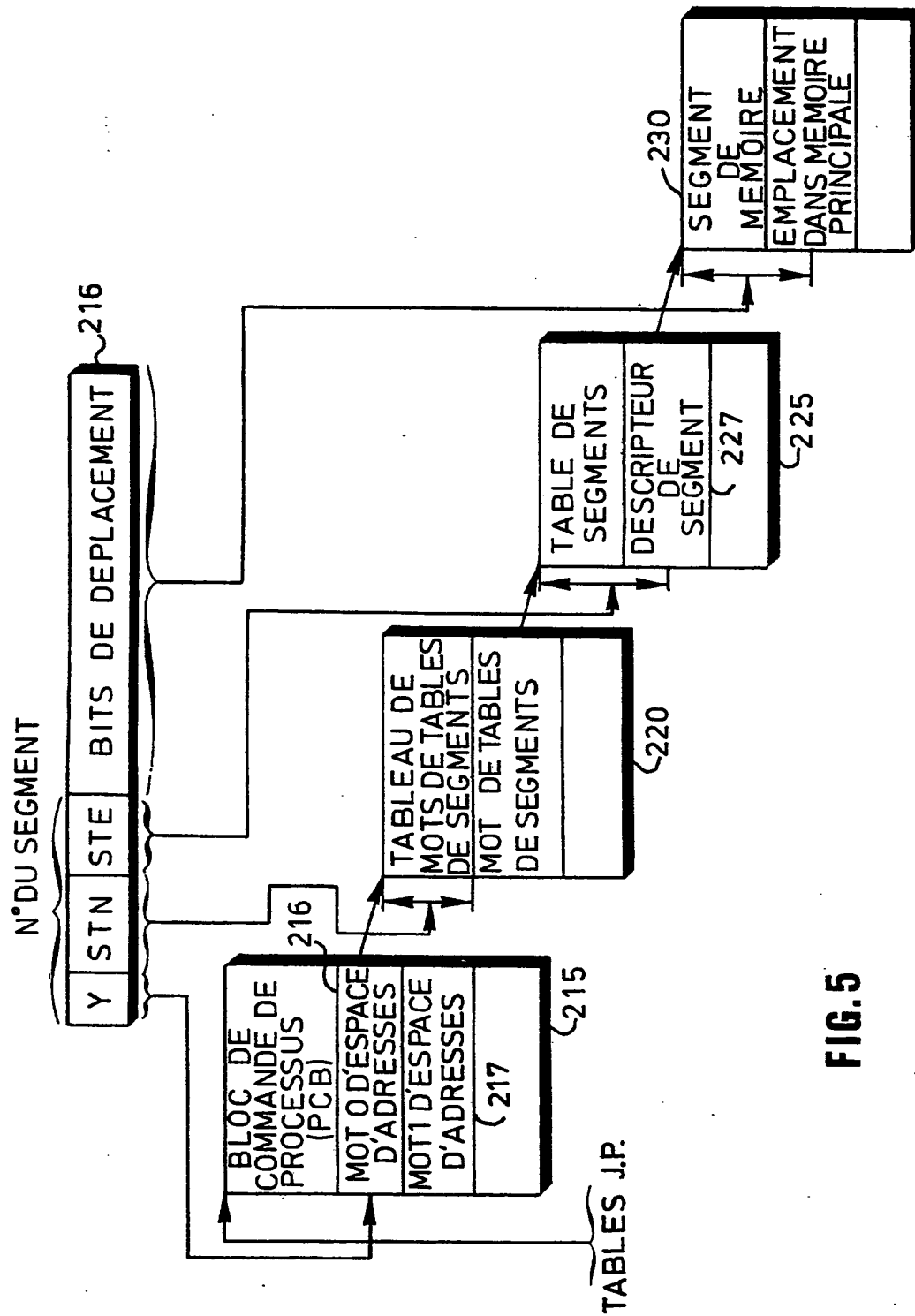


FIG. 5

FIG.6

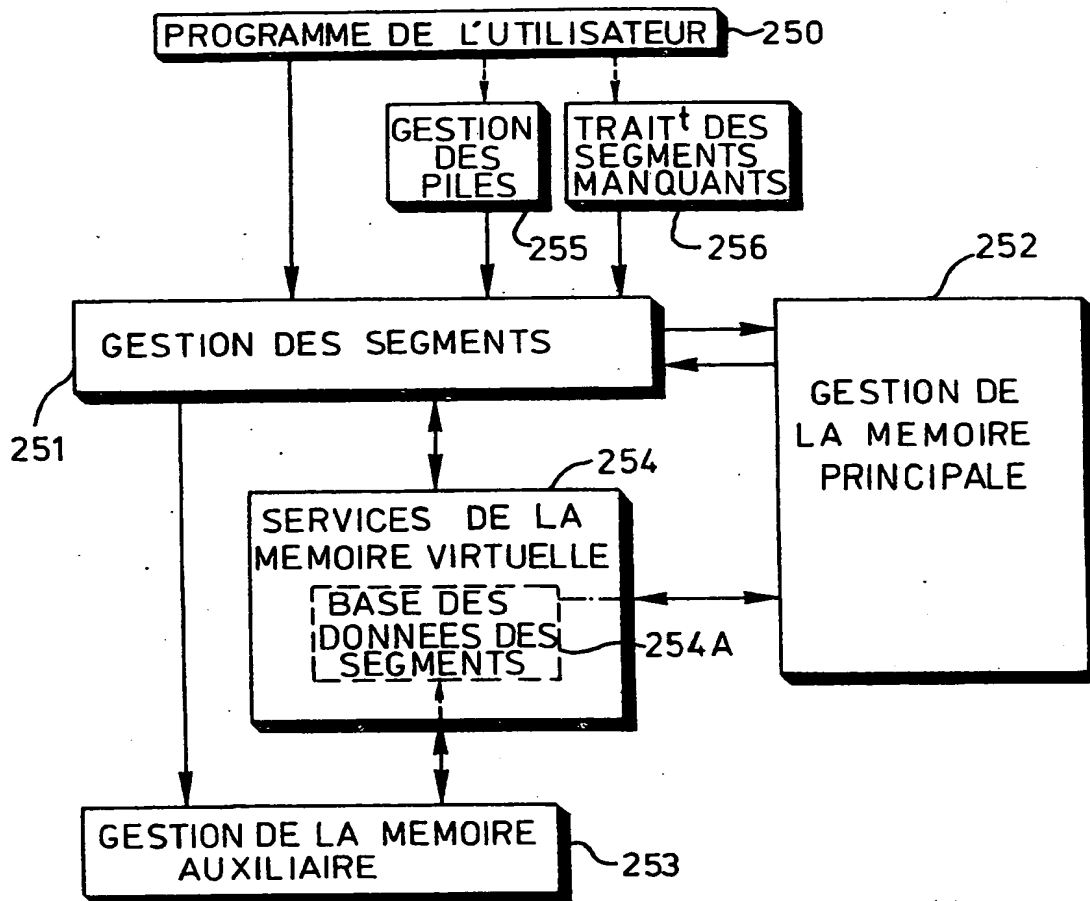
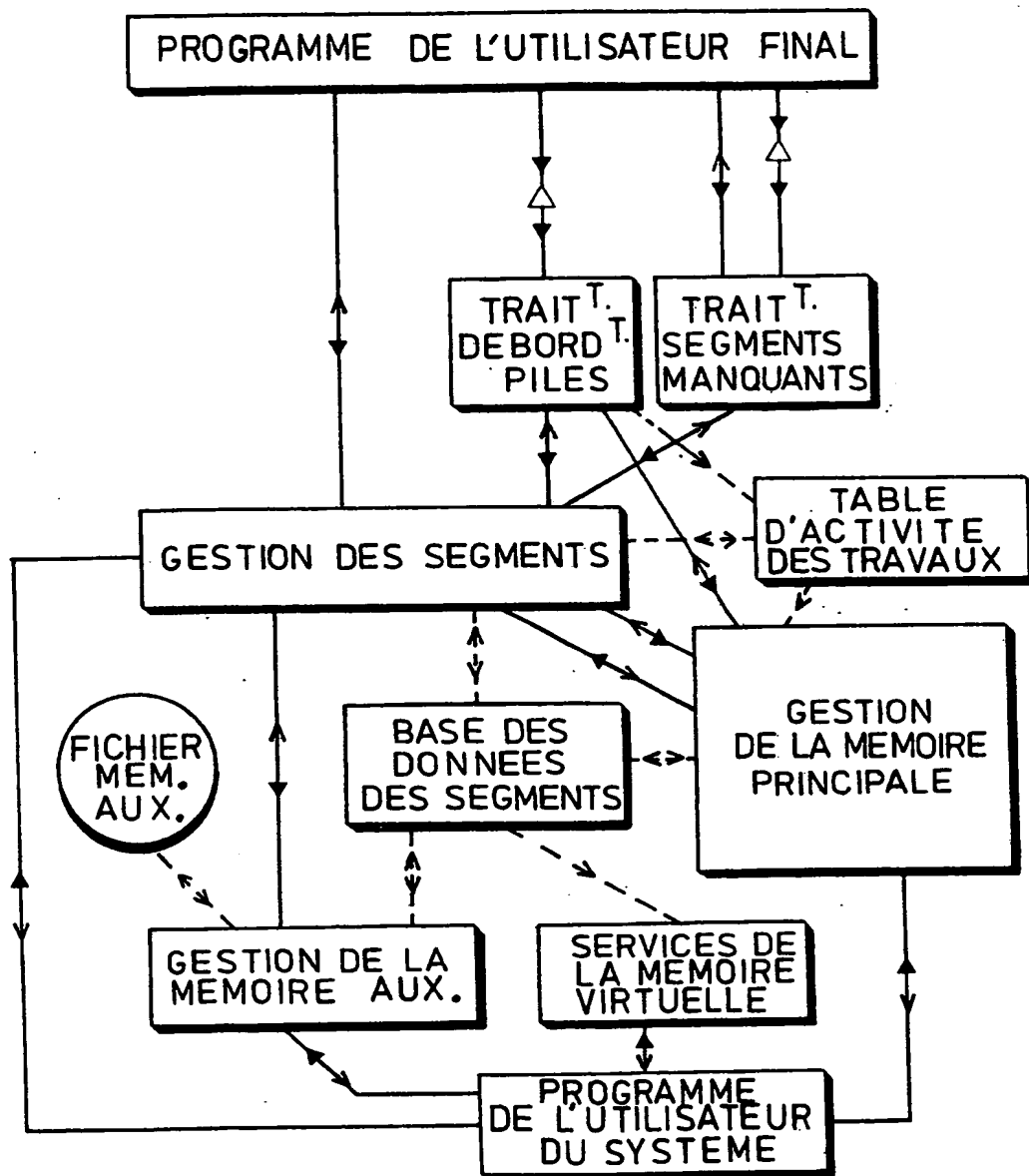


FIG. 7A

APPELEES DEMANDE- RESSES	SLDI	SUNLDI	SSIZEI	AUM	ACM	CUM	RMM	ABSM	MVMAIN	MVBACK	RBSM
SGLD	x			x					x		
SGUNLD							x	x		x	
SGSIZE	x			x	x	x	x				
SGATTR	x										
SDAVL											
SDUNVL											
RELV							x				x
ISDCRT											
ISDDEL											
ISDMOD											
SGMOVE	x										
LDUNLD	x	x									
VMMBUF											
VMMREL											
ISDGIL											
RPSP											
VMEXTR											
SLDI				x					x		
SUNLDI							x	x		x	
SSIZEI	x			x		x					
AUM		x									
CUM		x		x			x				
ACM											
RMM											
ABSM											
RBSM											
MVMAIN											
MVBACK											
MISSGHDL	x										
STKOVHDL	x		x	x							

FIG. 7B



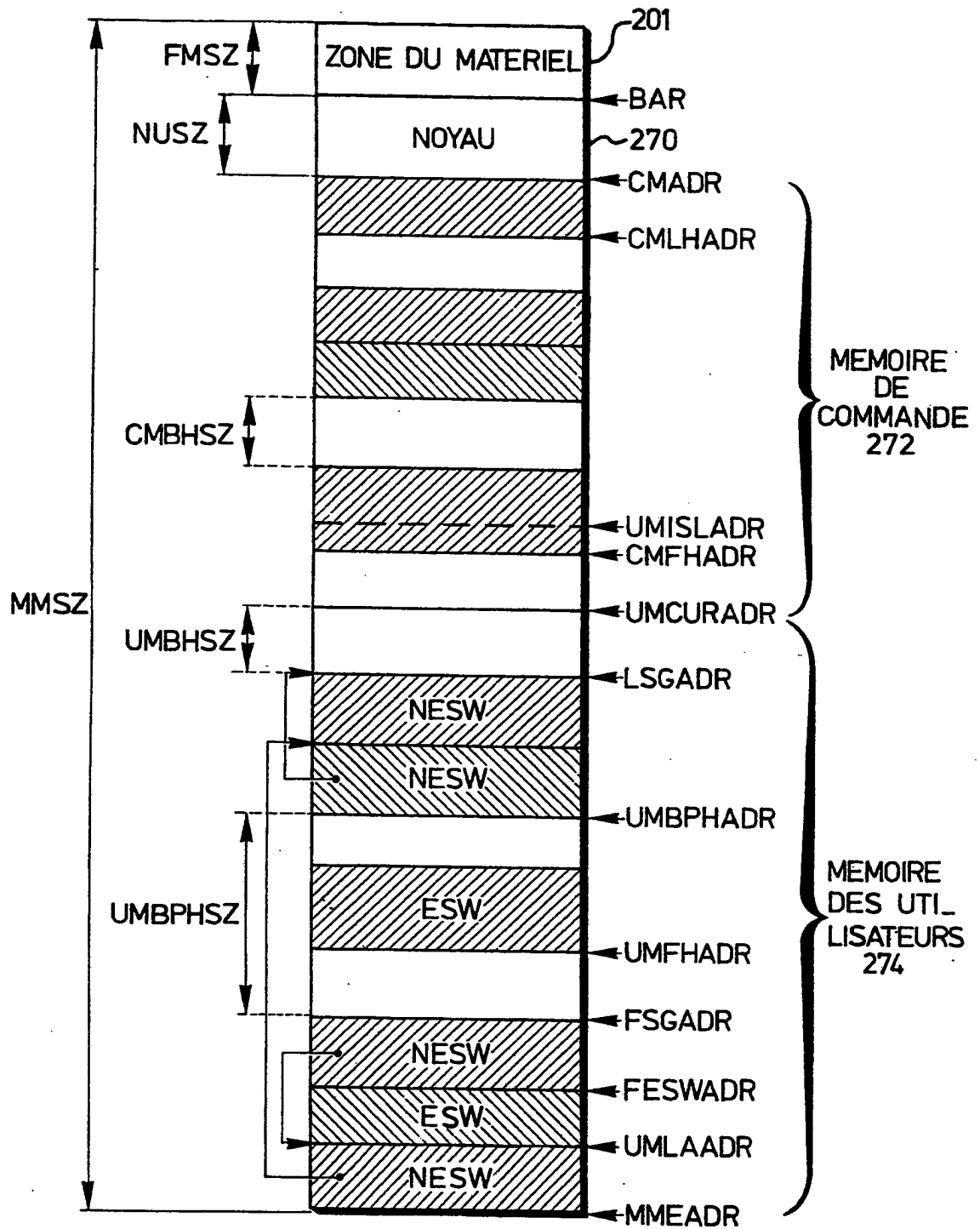
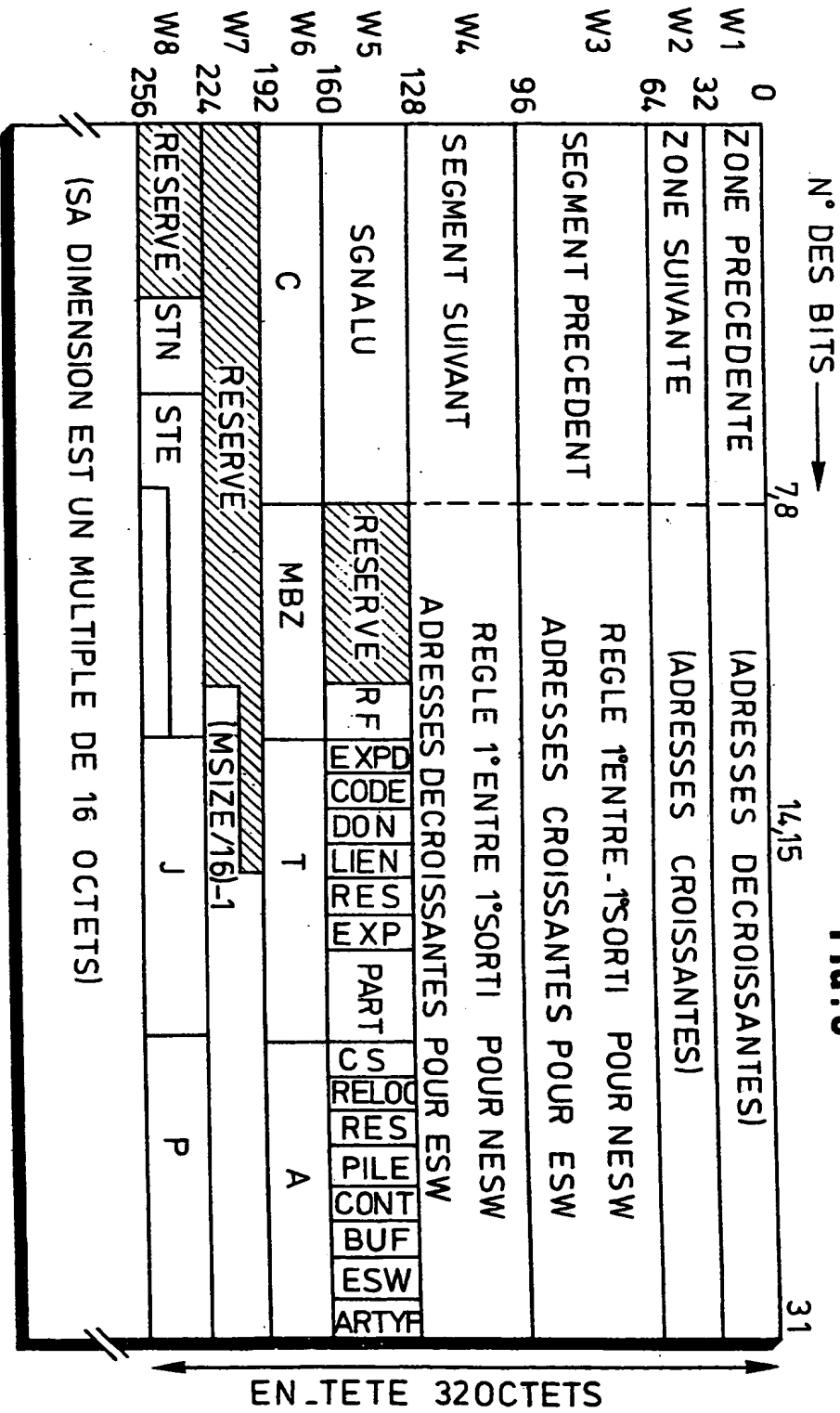


FIG. 8

FIG. 9



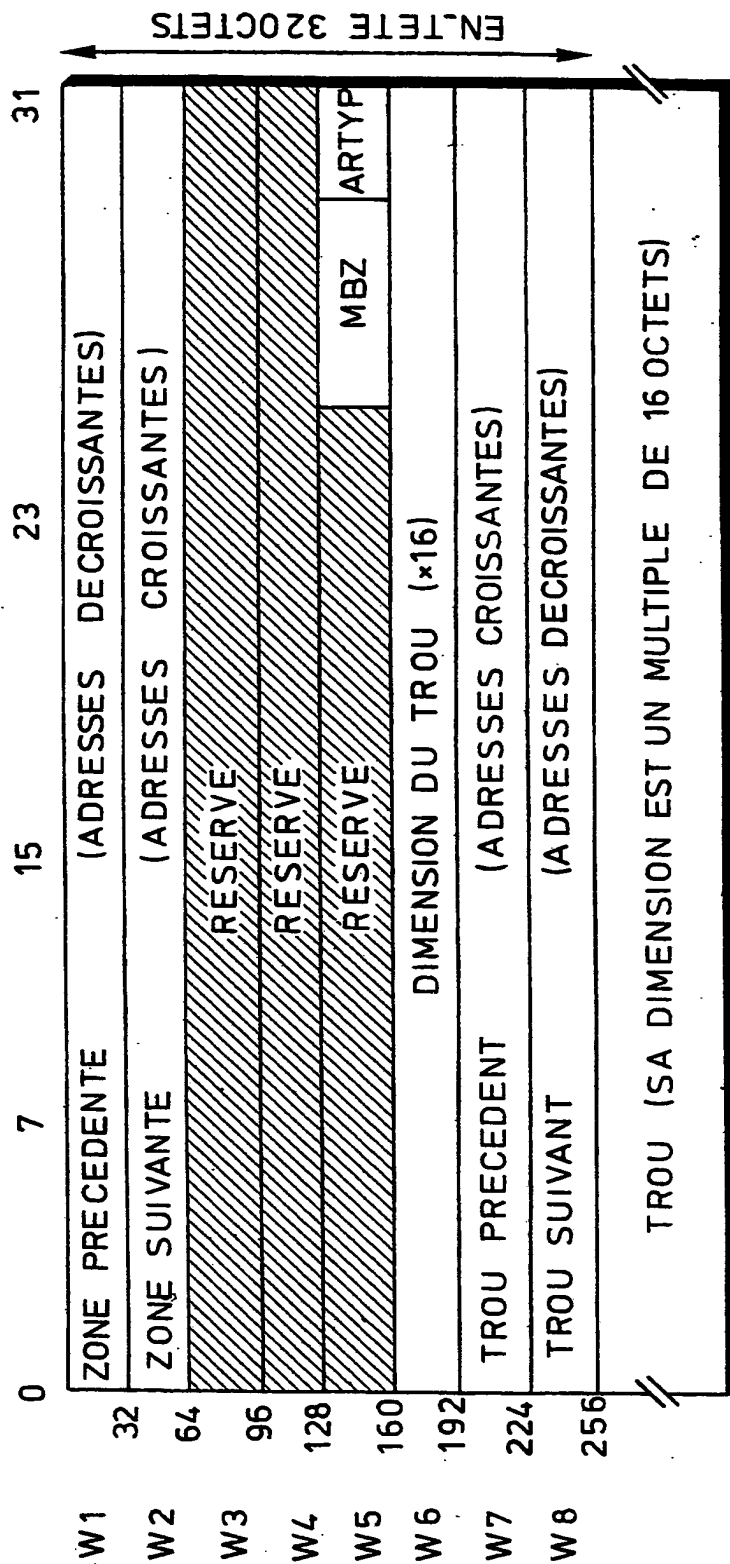


FIG.10

0 1 2 3 4

FIG. 12

0 1 2 3 4 5 6 7 8

P	I	A	U	W	G	S	ADRESSE ABSOLUE DU SEGMENT
ZONES D'ACCES							DIMENSION DU SEGMENT

COMBIN. DES ZONES	P	I	U	W
IDS	1	1	-	-
NESW	1	0	X	X
ESW	0	1	1	X
MANQUANT	0	0	0	0
FANTOME	0	0	0	1
INSTABLE	0	1	0	X
VACANT	0	0	1	0
RESERVE	0	0	1	1

DSD

X = 00U1

FIG. 13

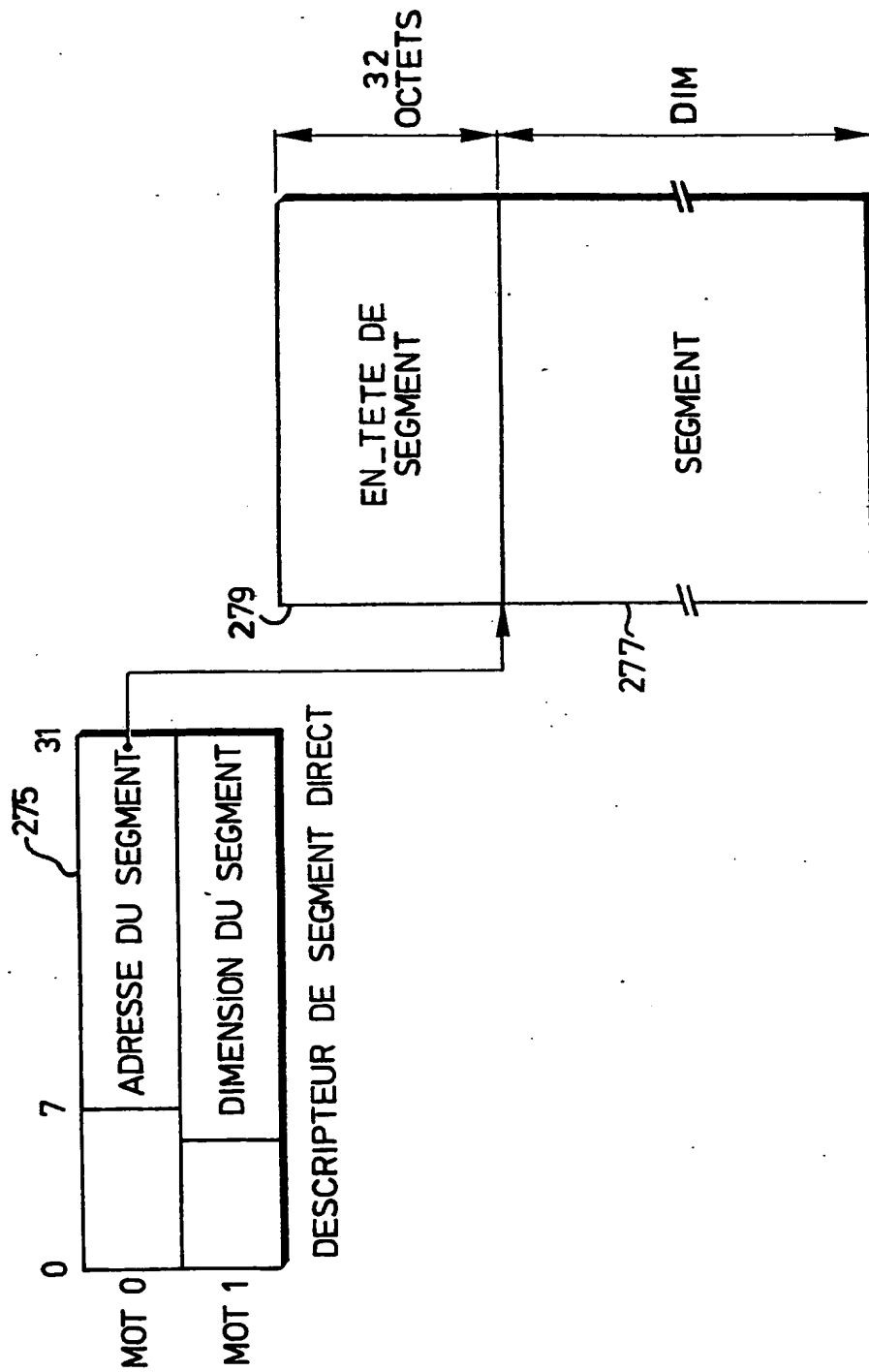


FIG. 14

FIG. 15B

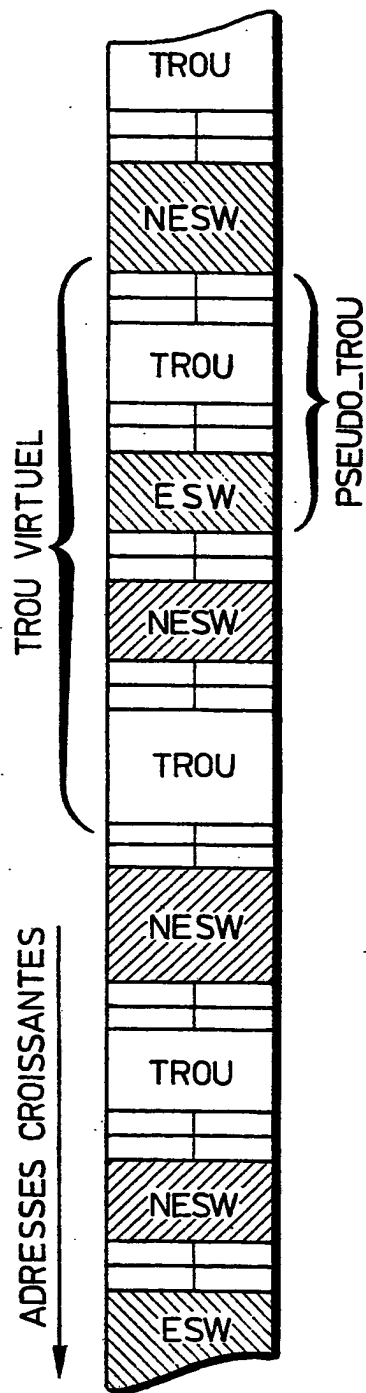


FIG. 15A

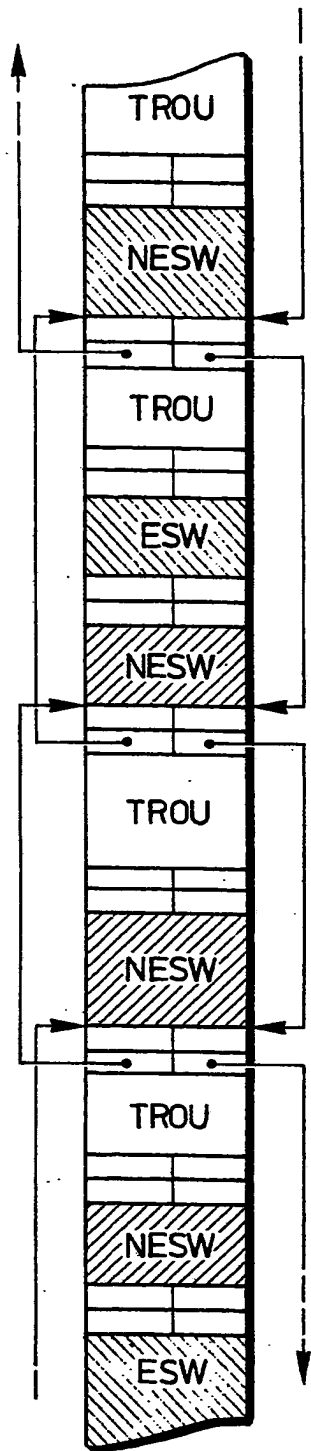
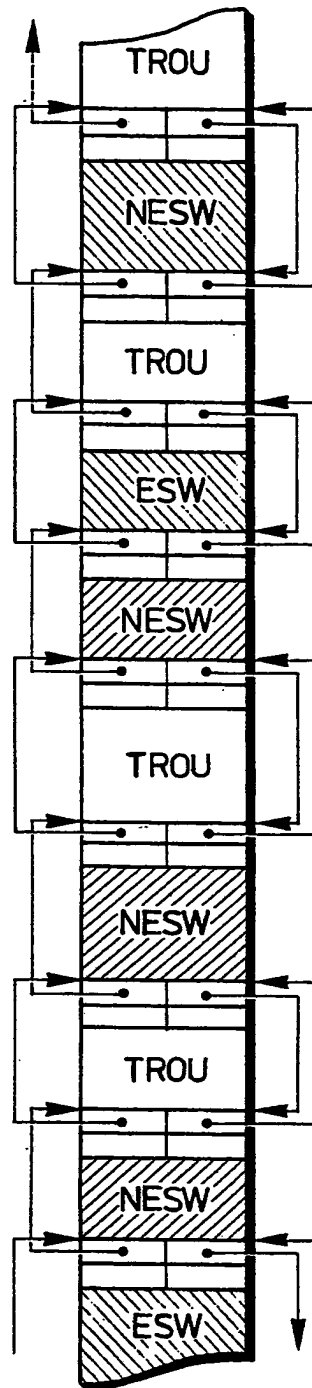


FIG. 15C

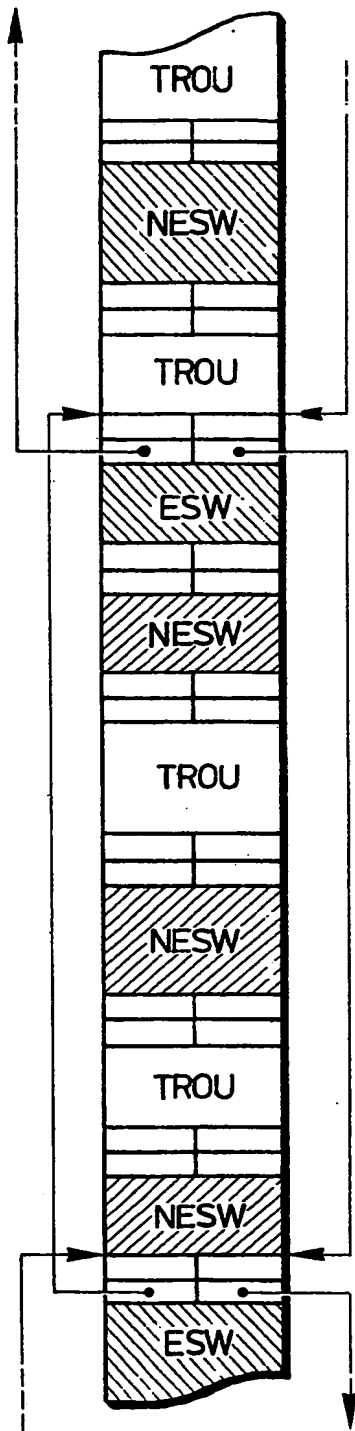


FIG. 15D

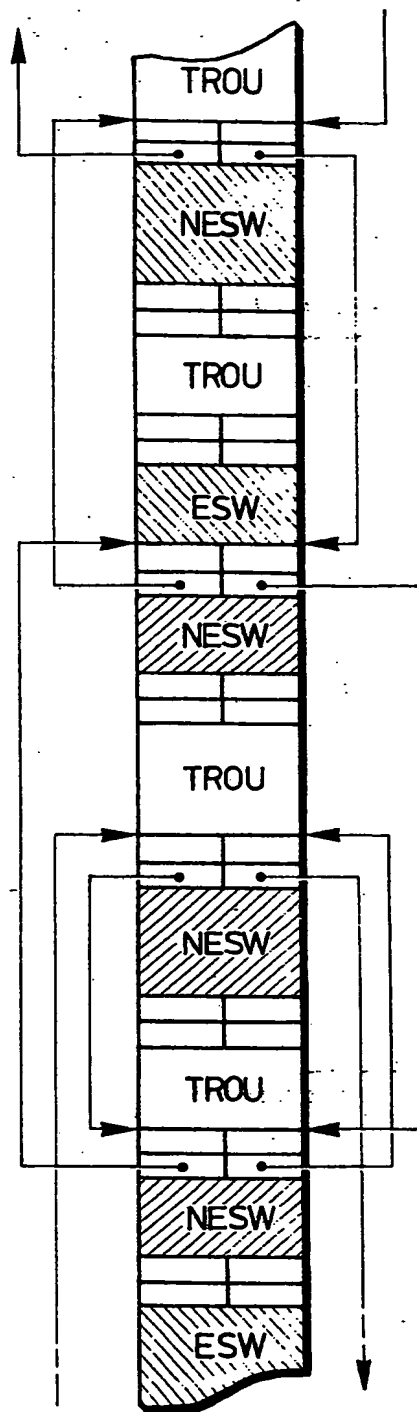
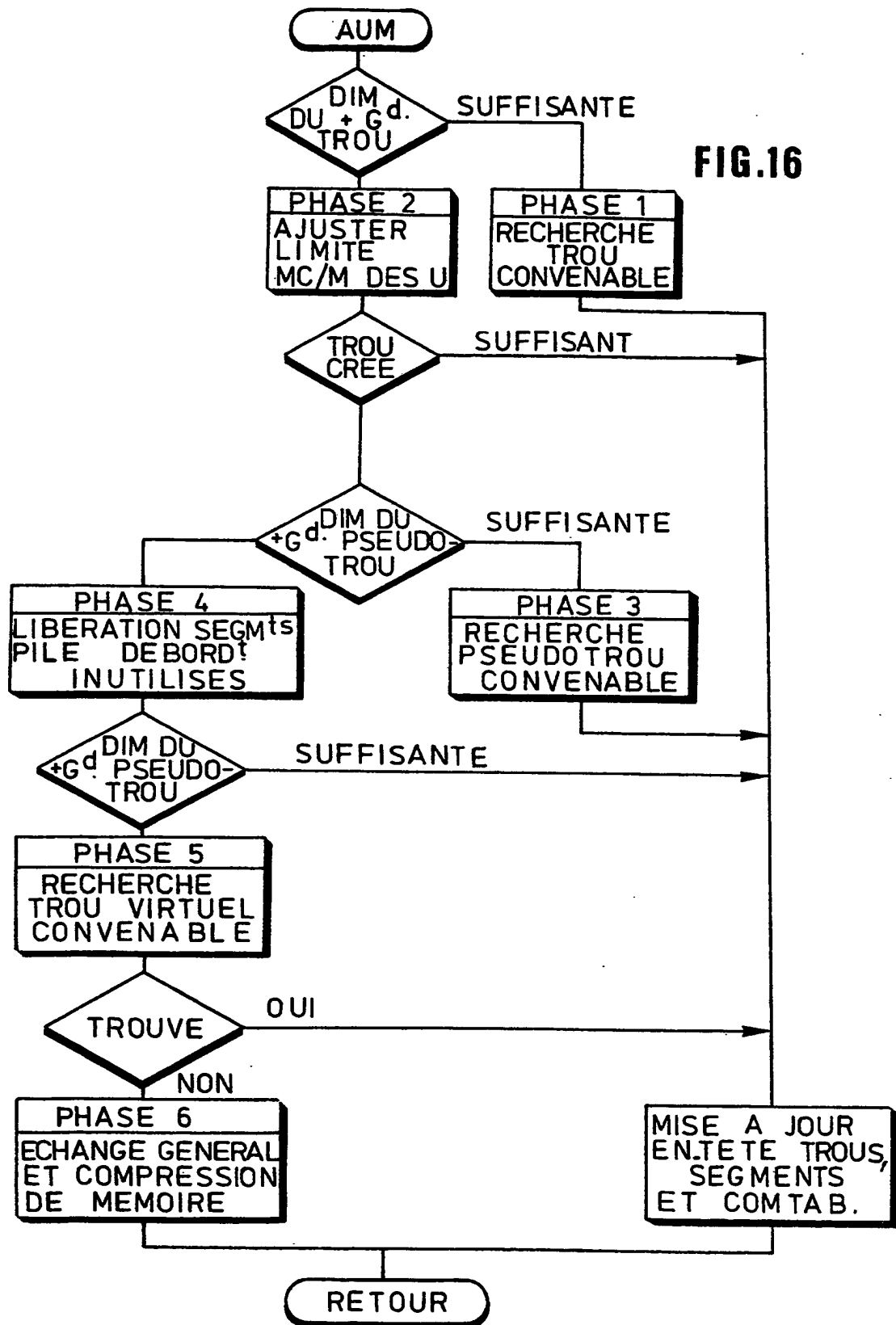


FIG. 15E

FIG.16



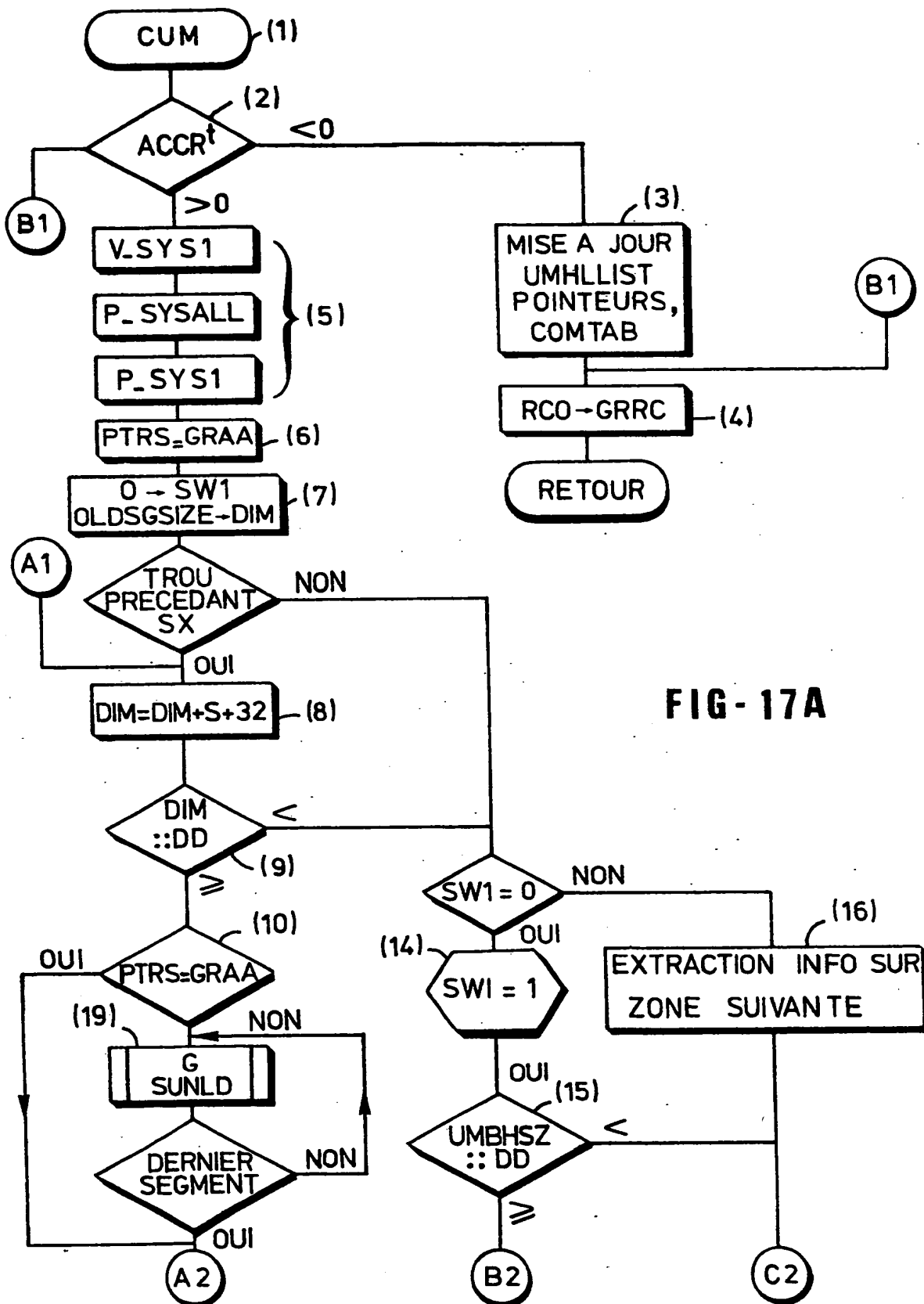
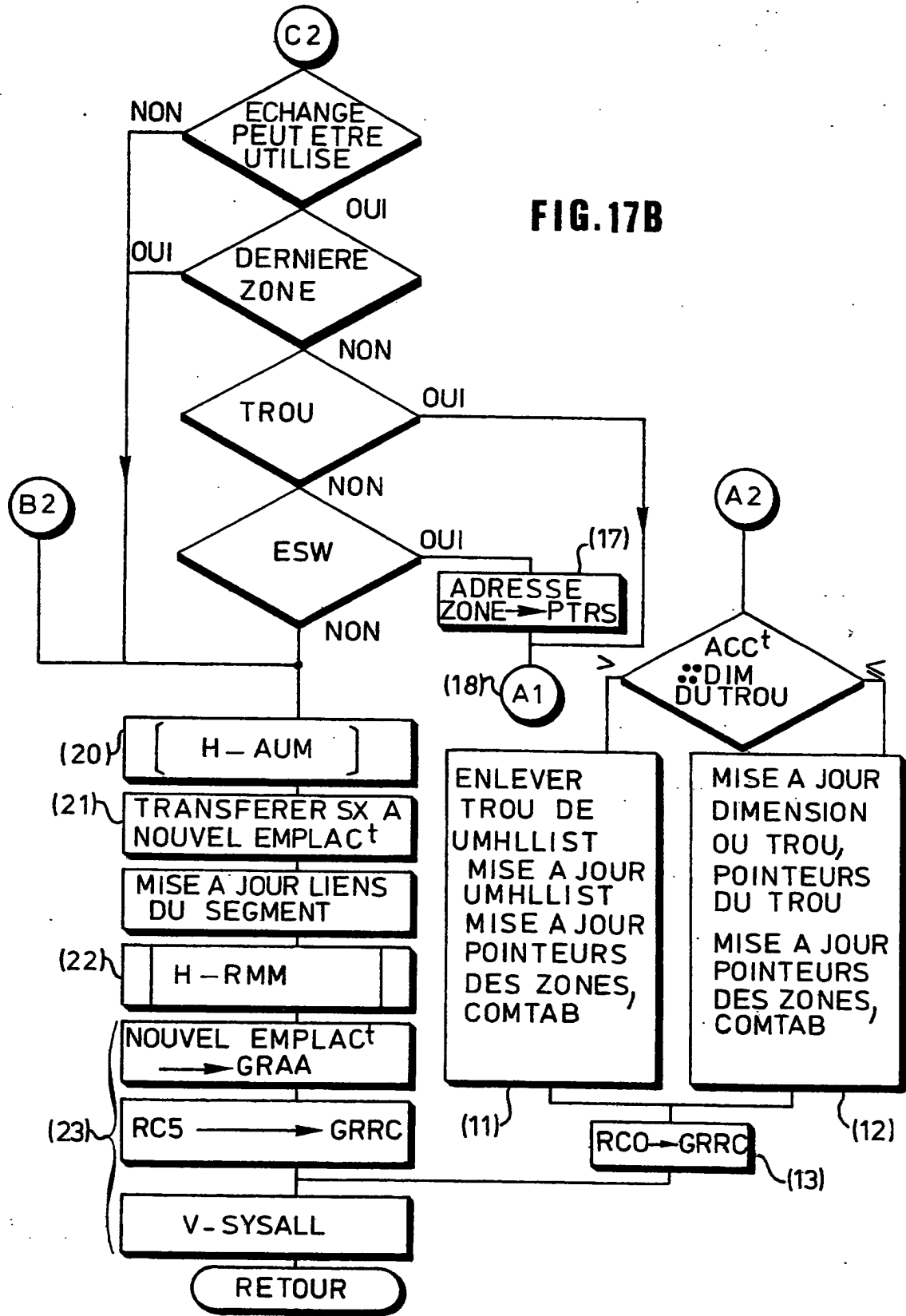
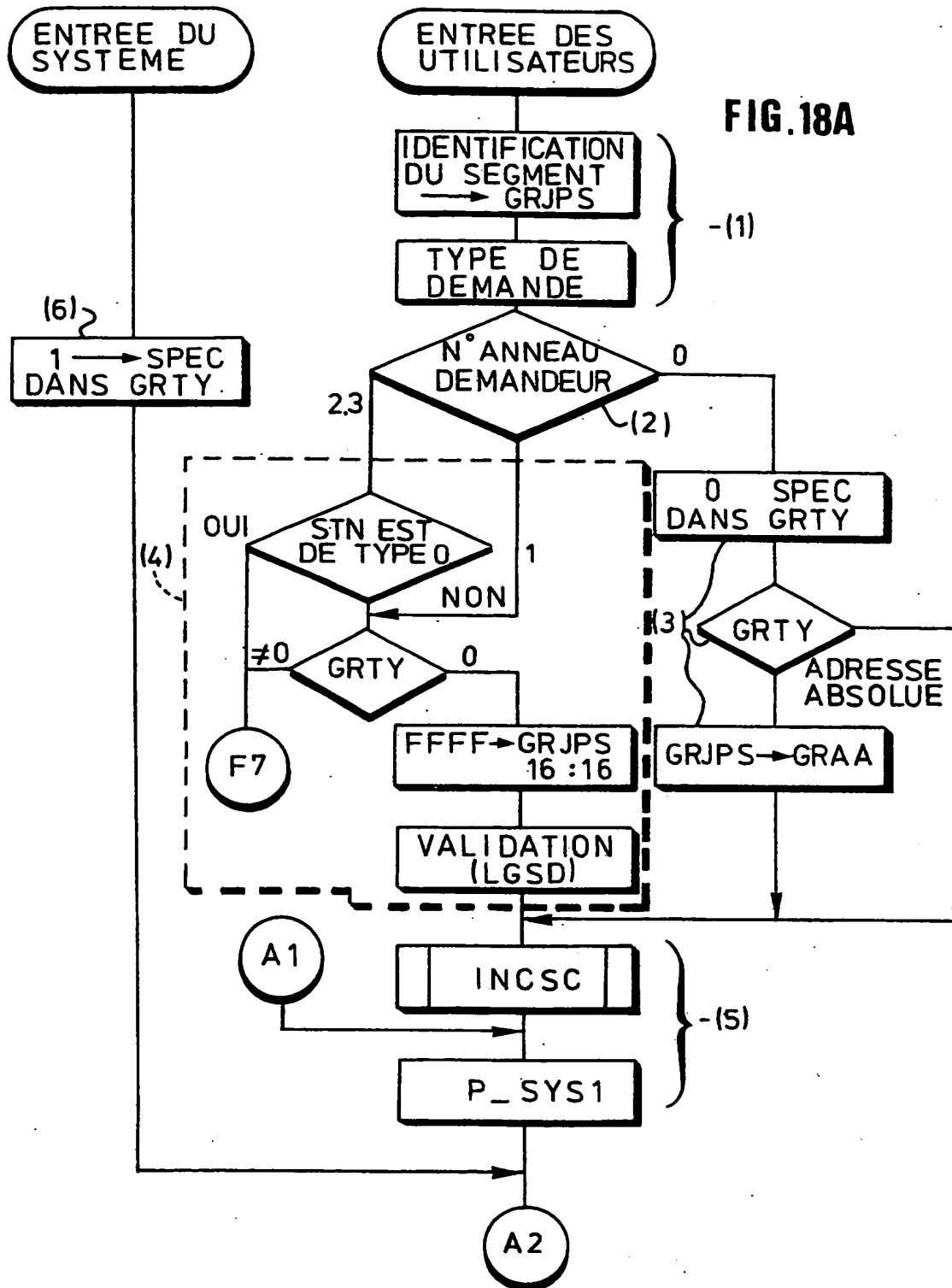
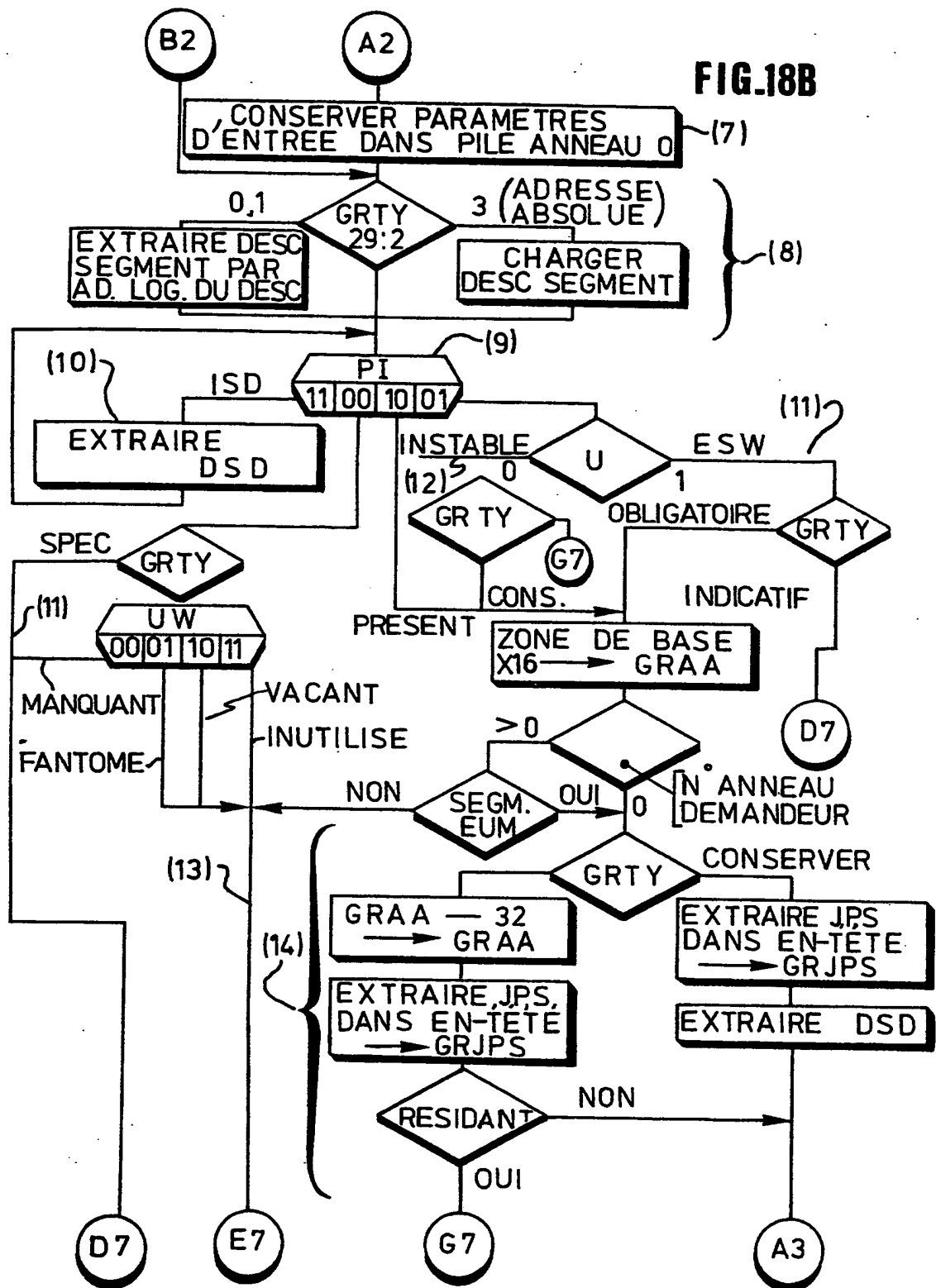


FIG-17A

FIG.17B







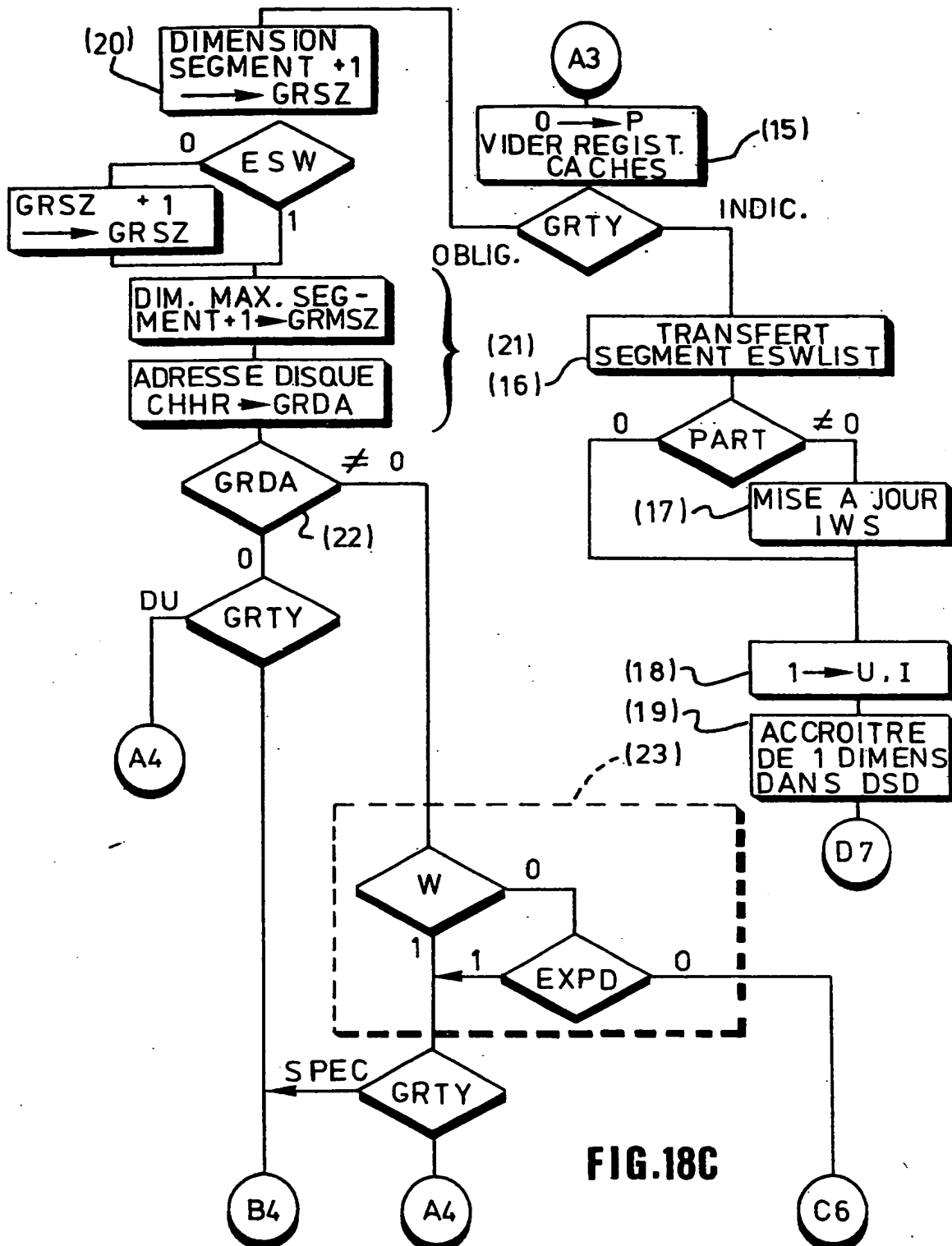
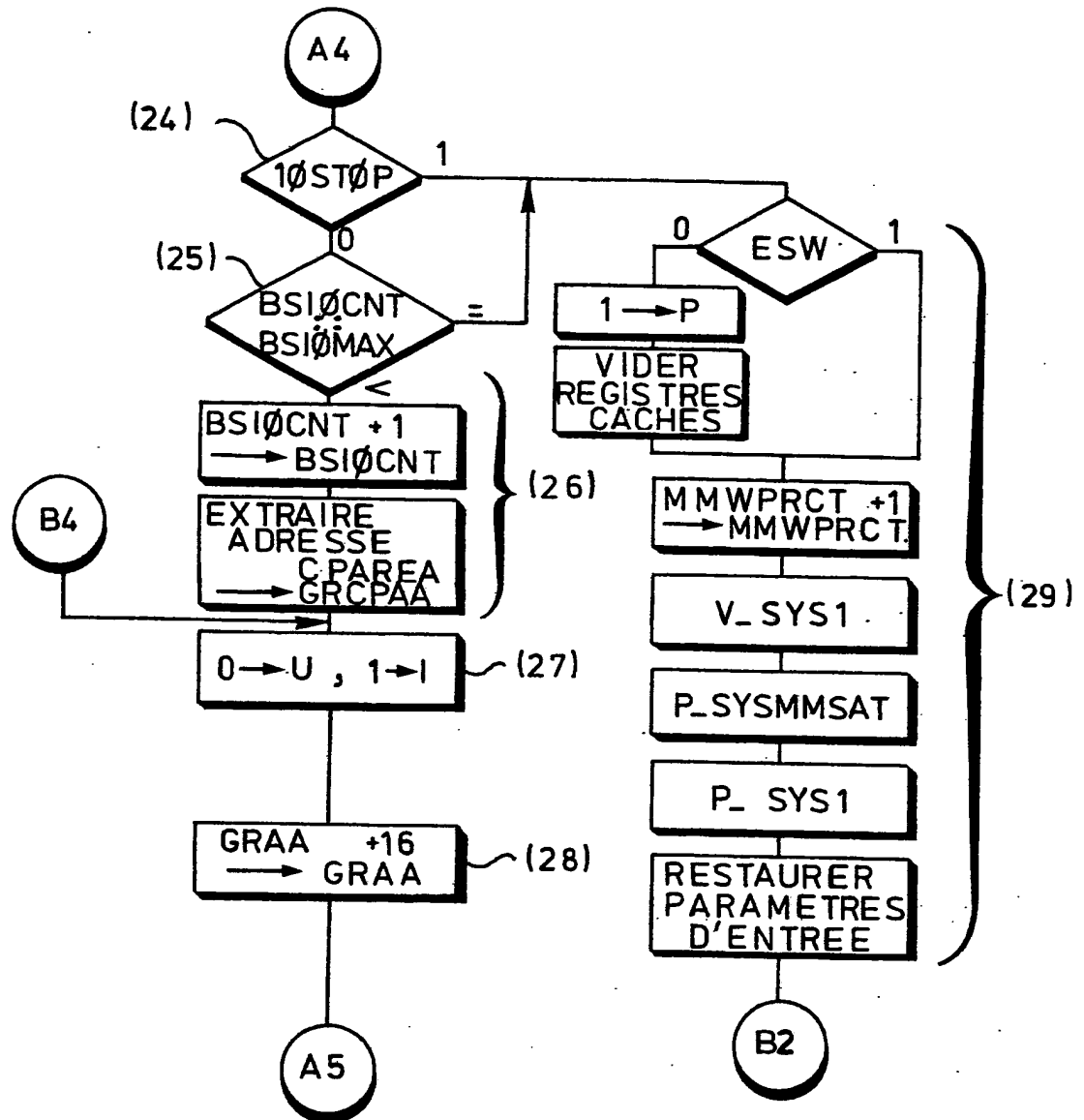
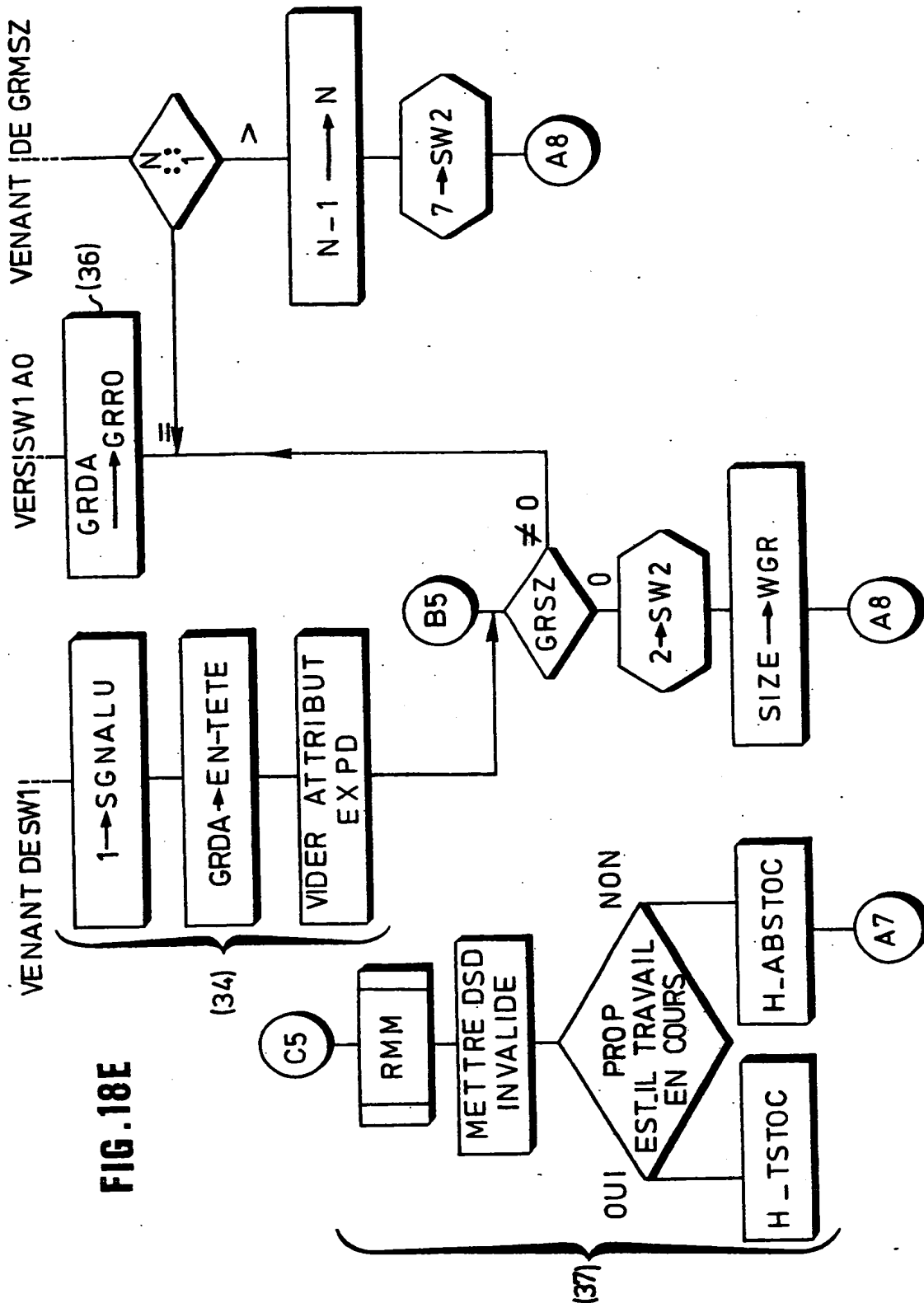
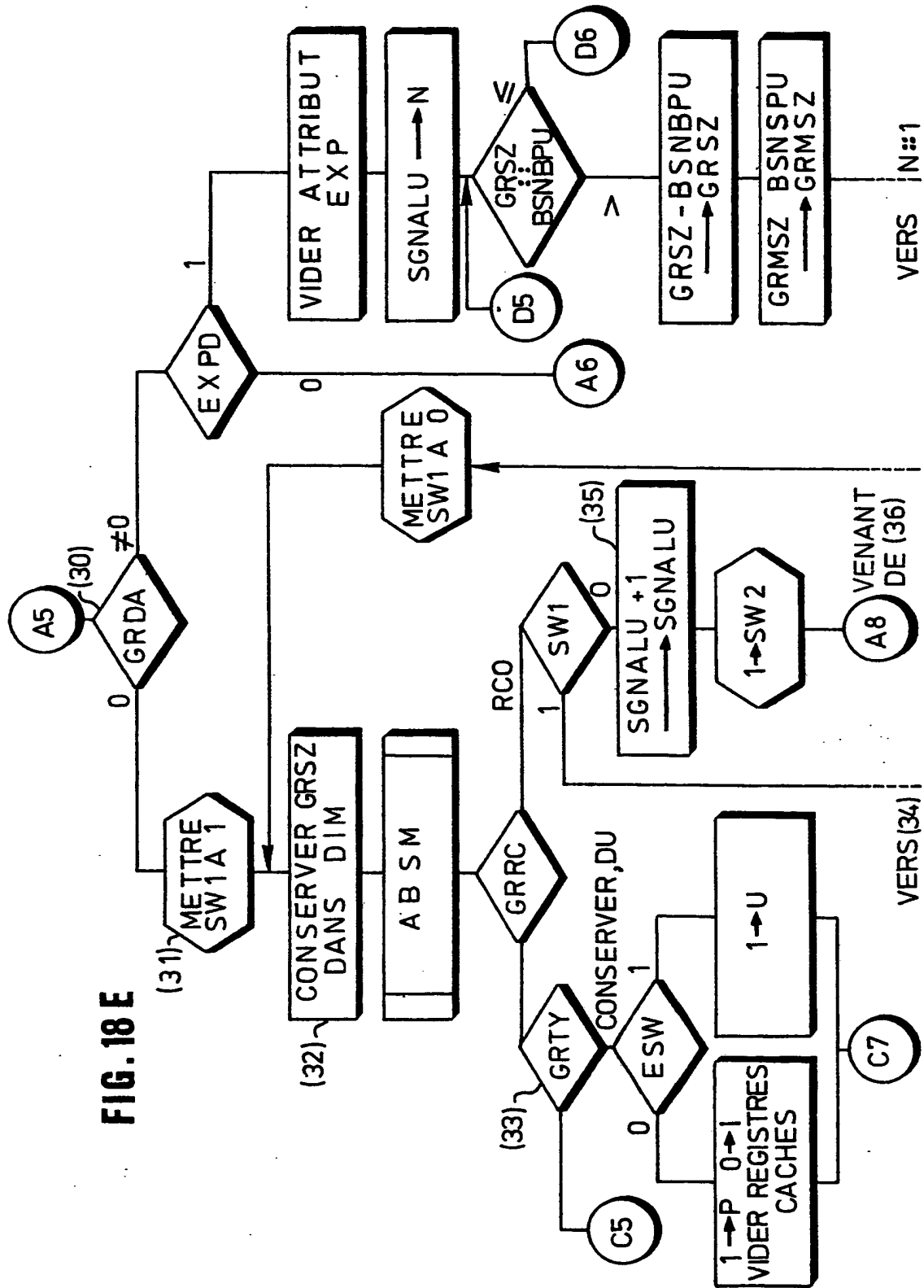


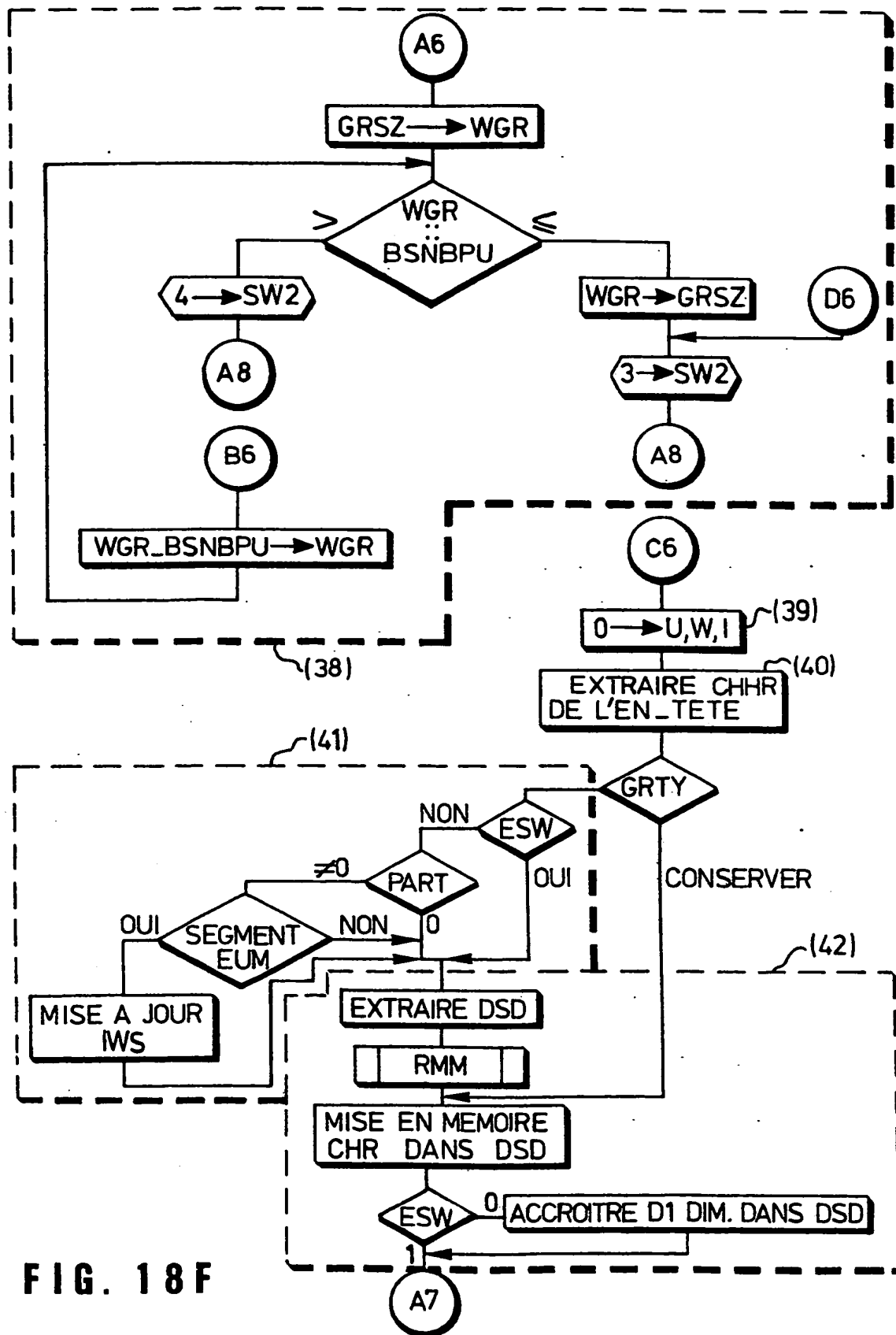
FIG.18C

FIG.18D









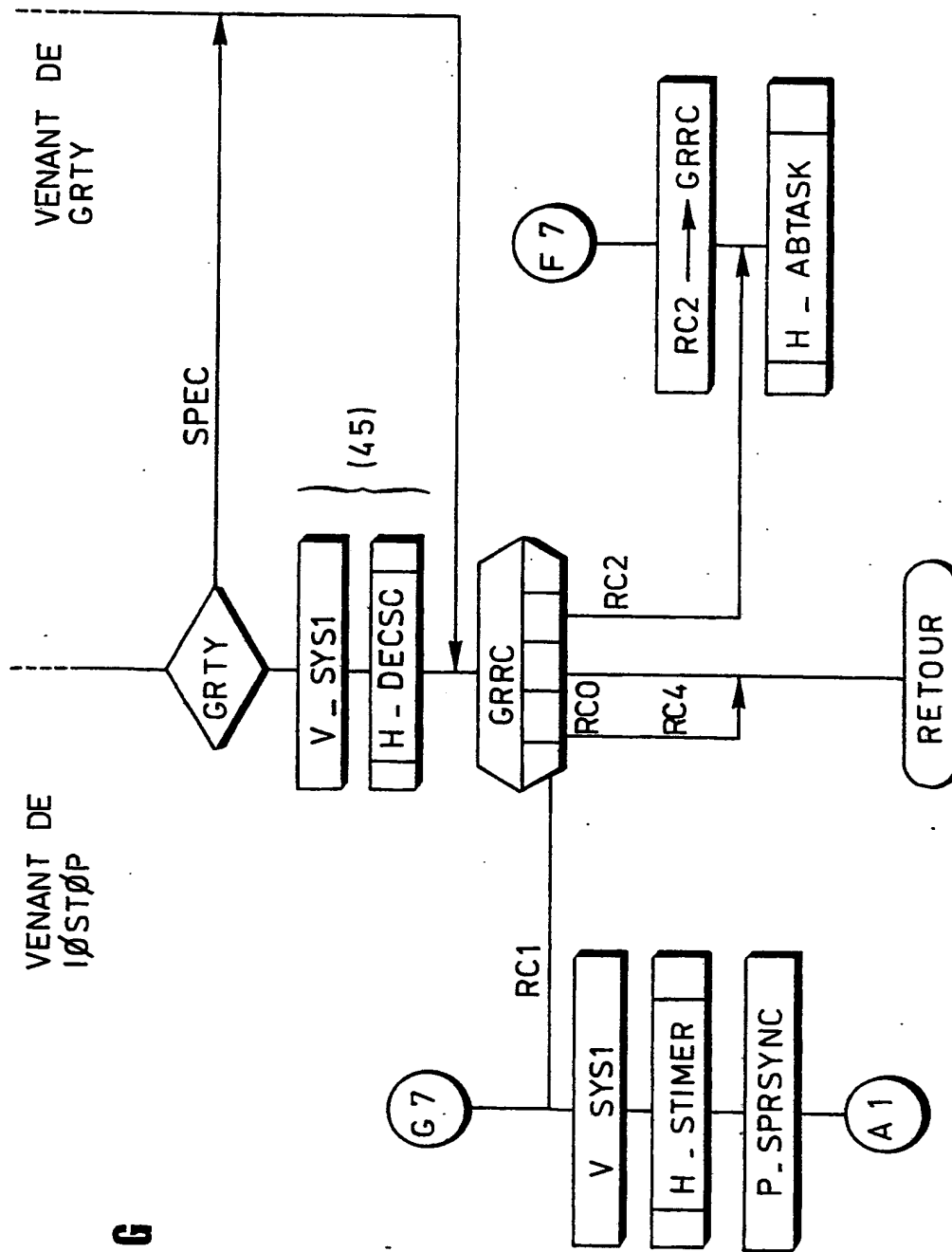


FIG. 18G

FIG. 18G

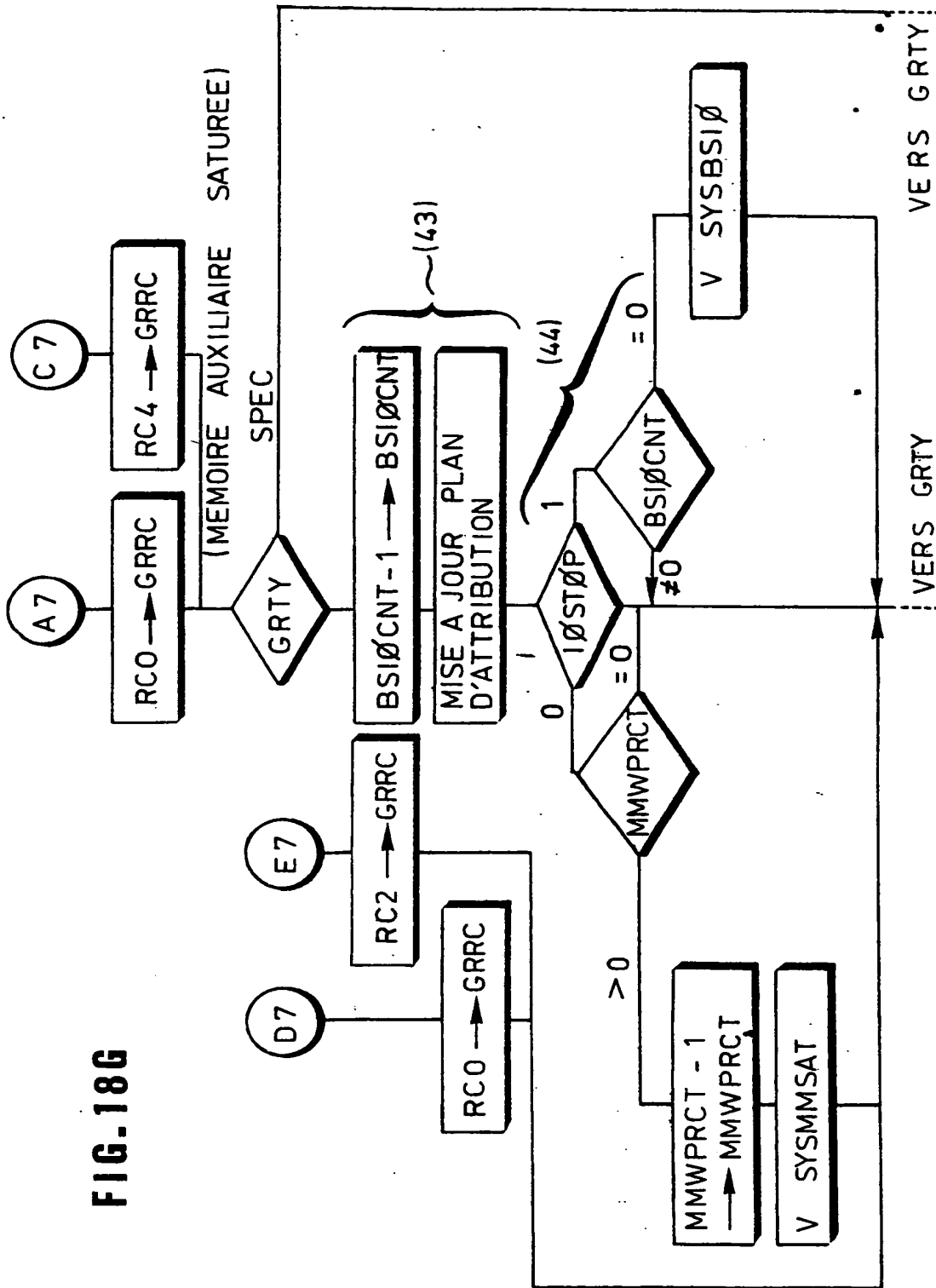


FIG. 18H

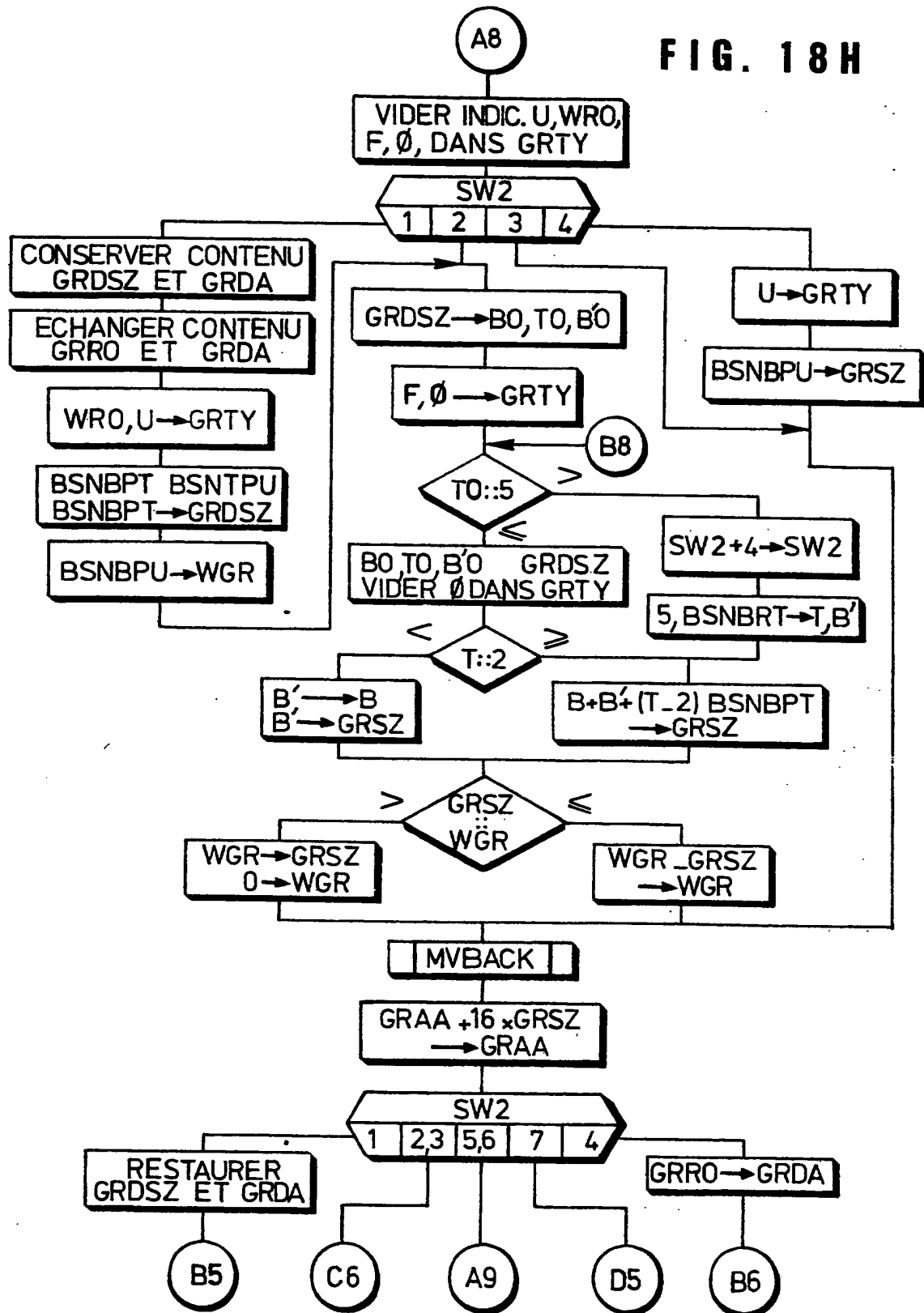


FIG. 18 I

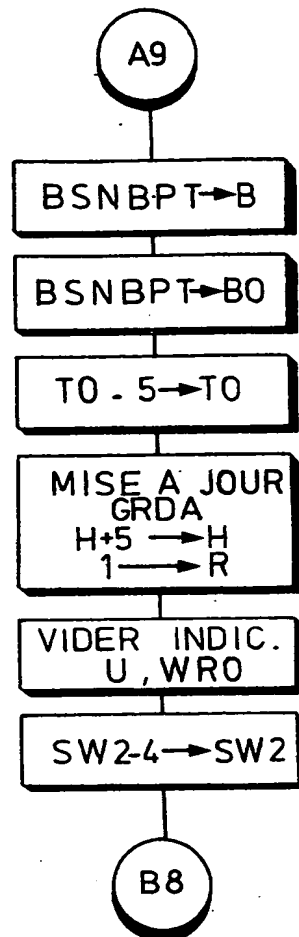
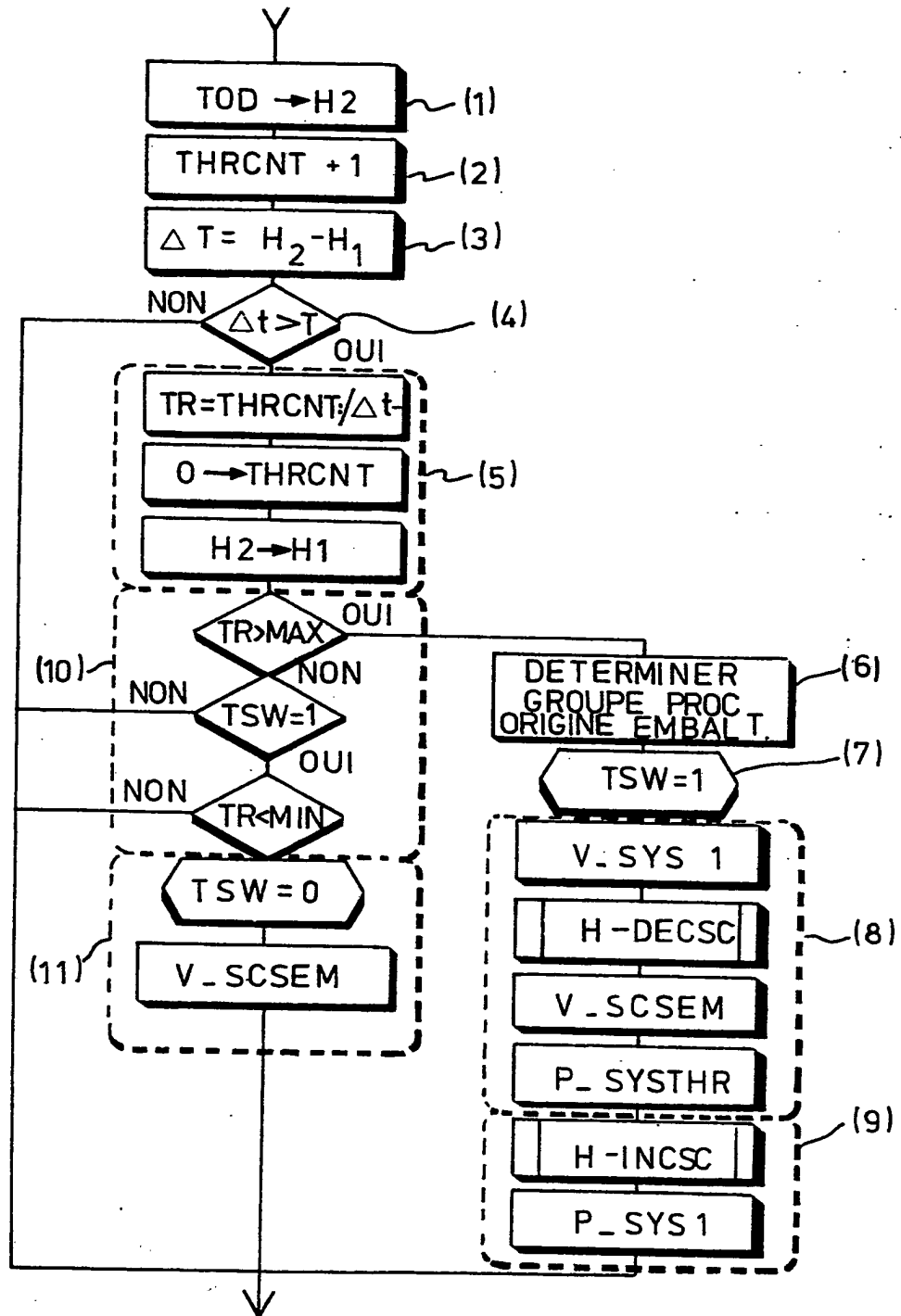


FIG.19



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)